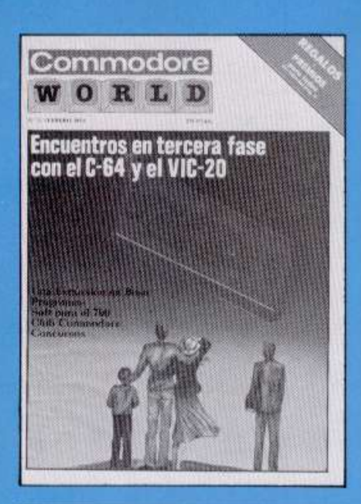
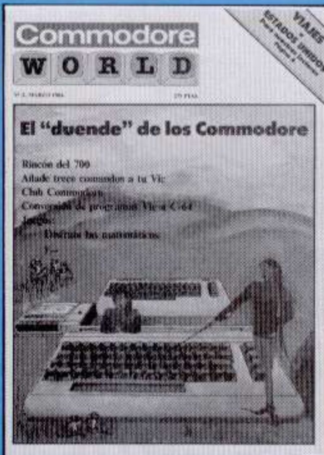
Commodore WORLD

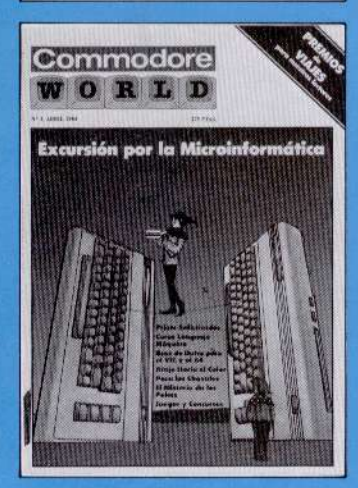
Nº 7 - SEPTIEMBRE 1984 315 PTAS.

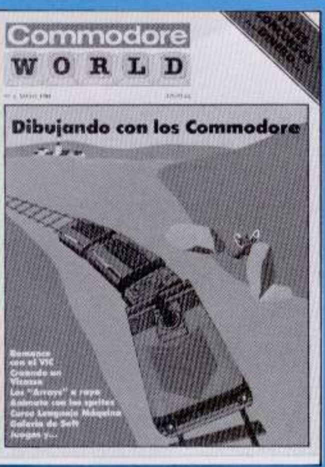
ESTUDIA NUESTROS CONCURSOS PERMANENTES - Pág. 4











"Commodore World"
aparte de venderse generalmente en kioscos, se encuentra asimismo a la venta en los siguientes distribuidores de Commodore y librerías.

ALICANTE
- Casa Wagner. c/ Juna Carlos I, 37. Elda. Teléfono: (965) 39 03 96.
- Ferretería Progreso. c/ General Jordana, 28. Elda. Teléfono: (965) 38 11 45.

BADAJOZ
- Control Sistemas. Avda. Santa Marina, 25A. Teléfono: (924) 25 88 00.

BARCELONA
- Computer Service. Avda. Abad Marcet, 325. Tarrasa. Teléfono: (93) 788 63 77.
- Computerhard. c/ s. Jaime, 48. Granollers. Teléfono: (93) 870 09 19.
- Gadesa. Les Valls, 12-14. Sabadell. Teléfono: (93) 725 25 43.
- Librería Emilia Pérez Radua. c/ Mayor, 35. Castellar del Vallés. Teléfono: (93) 714 89 51.
- Librería Michel. Ronda Guinardo, 1. Sardañola.
- Mozart. c/ Jaime I, 145. Mollet. Teléfono: (93) 593 75 01.
- Novo Digit. C/Aragón, 472. Telf.: 246 27 75
- Sonimóvil. c/ Alcalde Armengou, 53. Manresa. Teléfono: (93) 873 78 17.
- Tronic. Bigay, 11-13. Teléf.: 212 85 96 (93)

BILBAO
- Bilbomicro SA. c/ Aureliano Valle, 7. Bilbao-10. Teléfono: (94) 443 43 51.

BURGOS
- E.I.S.A. c/ Madrid, 4. Teléfono: (947) 20 46 24.

CADIZ
- Video Computer. Comandante Gómez Ortega, 59. Algeciras. Teléfono: 653 902 (956)

CANARIAS
- Relax. Rambla de Pulido, 85. Santa Cruz de Tenerife. Teléfono: (922) 28 37 05.

CIUDAD REAL
- CISA. c/ D. Víctor, 15. Tomelloso. Teléfonos: (926) 51 00 07 - 51 15 72.
- Electrónica Turrillo. c/ Pedrera Baja, 7. Teléfono: (926) 22 38 67.

CORUÑA, LA
- Cetronic, S.L. c/ Palomar, 22 Bajo. Teléfono: (981) 27 26 54.
- Photo Copy. c/ Teresa Herrera, 9. Teléfono: (981) 21 34 21.
- Sanlusa, S.L. San Luis, 46 al 50. Teléfono: (981) 23 07 49.

GERONA
- Digit Informática. c/ Avda. 11 Setembre, 7. Olot. Teléfono: (972) 26 94 01.
- Microchip. C/Aigua, 3. Olot. Teléf. 263 663
- Regiscompte SA. c/ Emilio Grahit, 17 Bis. Teléfono (972) 21 99 88

HUELVA
- Computerlog, S.A. c/ Tendaleras, 15. Teléfono: (955) 25 81 99.

IBIZA
- Kelson. Plaza España, S/N. San Antonio Abad. Teléfono: (971) 34 13 09.

LEON
- MicroBierzo. c/ Carlos I, 2. Teléfono: (987) 41 74 21. (Ponferrada)

MADRID
- Chips + Tips. c/ Puerto Rico, 21-23.
- Librería García Peña. c/ Cavanilles, 52.
- País de los Microordenadores. Bravo Murillo, 18. Teléfonos: (91) 446 33 17 - 446 37 12.

MALAGA
- Informática Martínez S.A. c/ Cristo de la Epidemia, 90. Teléfono: (952) 26 15 60.

MURCIA
- Procoinsa. Ronda Norte, 27. Teléfono: (968) 23 94 49.

PAMPLONA
- Microordenadores Ramar. c/ Navarro Villoslada, 7. Teléfono: (948) 23 72 80.

SALAMANCA
- Compusoft. c/ Arco, 1. Teléfono: (923) 21 59 93.

SANTANDER
- Librería Hernández. San Francisco, 15. Teléfono: (942) 22 53 30.

SANTIAGO DE COMPOSTELA
- GestyComputer. Avda. Romero Donallo, 25. Teléfono: (981) 59 87 54.

SEVILLA
- Papelería Mora. c/ Santa Cruz, 5. Ecija Teléfono: (954) 83 14 80.

TARRAGONA
- Oficomplet. Plaza de la Cinta, 6. Tortosa. Teléfono: (977) 44 14 50.
- Comercial Informática de Tarragona SA. calle Gasómetro, 20. Teléfono: (977) 23 08 53.

VALLADOLID
- Chips + Tips, S. A. c/ Juan de Juni, 3. Teléfono: (983) 33 40 00.
- Kent Electrónica C/Duque de la Victoria, 13. tienda, 15. Teléfono: (983) 47 58 07

ZARAGOZA
- ADA Computer. Centro Independencia. Paseo Independencia, 24-26. Teléfono: (976) 29 85 62.

TIRADA DE COMMODORE WORLD
Certificación de 28.000 ejemplares

Según certificado firmado por la auditoría internacional DELOITTE HASKINS SELLS, S. A. el 5 de marzo de 1984, la tirada de Commodore World es de 28.000 ejemplares. Este certificado se encuentra en las oficinas de S.I.M.S.A. para comprobación de cualquier persona o entidad que lo desee.

Commodore WORLD

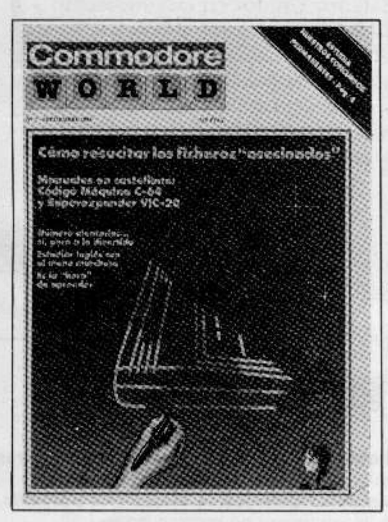
Commodore World
está publicado por SIMSA
y la colaboración
de todos nuestros lectores

EQUIPO

Manuel AMADO; Nieves CHESA; José Luis ERRAZQUIN; Miguel A. HERMOSELL; María LOPEZ; Juan MARQUEZ Juan MARTINEZ; Pere MASATS; Victoria MORALES; Rafael PARDO; Diego ROMERO; Albert SANGLAS; Jordi SASTRE; Valerie SHANKS...
...Y NUESTROS LECTORES

SIMSA
Coordinador María López
c/ Barquillo, 21-3º. Izda. - 28004
Madrid Teléf.: (91) 231 23 85/85
Delegación en Barcelona:
Sant Gervasi de Cassoles, 39 despacho
4 Barcelona-22.
Tel.: (93) 212 73 45/212 88 48

Colabora MEC-COMMODORE
con Club Commodore
Coordinador Pere Masats
Taquígrafo Serra, 7-5º
Barcelona-29
Tels.: (93) 250 51 03/02

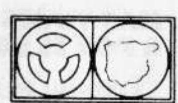


Imprime:
IBERDOS, S. A.
Germán Pérez Carrasco, 24.
Madrid-27
Depósito Legal: M-2944-1984

SUMARIO

PROXIMO NUMERO

- No te deshagas de tu VIC cuando compras un 64.
- Qué hacer con tu cassette para que entren esas cintas "rebeldes"
- Disk-O-64
- Pasaporte al mundo
- Viaje al centro de tu Commodore



Commodore World es miembro de CW Communications/Inc., el grupo editorial más grande y de mayor prestigio en el mundo en lo que se refiere al ámbito informático. Dicho grupo tiene a su cargo la edición de 52 publicaciones relacionadas con los ordenadores en 19 países. El grupo editorial está integrado por:
ALEMANIA: Computerwoche, Micro Computerwelt, PC Welt, Software Markt, CW Edition/Seminar, Computer Business, Run (Commodore). **ARABIA SAUDI:** Saudi Computerworld. **ARGENTINA:** Computerworld/Argentina. **AUSTRALIA:** Australasian Computerworld, Micro Computer Magazine, PC World. **BRASIL:** DataNews, Micro-Mundo. **DINAMARCA:** Computerworld/Danmark, Micro Verden. **ESPAÑA:** Computerworld/España. MicroSistemas, Commodore World. **ESTADOS UNIDOS:** Computerworld, Hot CoCo, InCider, InfoWorld, Microcomputing, PC World, 80-Micro, Mac World Micro Market Word, PC Jr. World Run (Commodore). **FINLANDIA:** Mikro. **FRANCIA:** Le Monde Informatique. **HOLANDA:** Computerworld Benelux, Micro/Info. **ITALIA:** Computerworld Italia. **JAPON:** Computerworld Japan, Perso Comworld, PC World. **MEXICO:** Computerworld/México, Compumundo. **NORUEGA:** Computerworld Norge, Mikro Data. **REINO UNIDO DE GRAN BRETAÑA:** Computer Management. Computer Business Europe. **REPUBLICA POPULAR DE CHINA:** China Computerworld. **SINGAPUR:** Asian Computerworld. **SUECIA:** ComputerSweden. MikroDatorn, Min Hemadator.

SUPER-INTERESANTISIMO

NOSOTROS MISMOS... (y vosotros como siempre)

Tenéis en vuestras manos el número de Septiembre, con el verano terminado y de vuelta a la vida normal. Sin embargo nosotros estamos escribiendo estas líneas el 30 de julio, con todo el calor del mundo y en onda de comenzar vacaciones en vez de volver de ellas, seguros de que tenemos un montón de cosas que decir, pero, con toda la sinceridad del mundo, nos sentimos incapaces de pensar coherentemente con 40º y la menta escapándosenos continuamente al placer de una buena piscina.
Por lo tanto, como recién venidos del verano, tampoco estáis con ganas de mucho "rollo" pasaré a tres cosas rápidas

Lo más importante es
CAMBIO DE DOMILICIO
Hemos crecido, y crecido y nos hemos quedado pequeños de espacio e instalaciones. La nueva casa de Barquillo seguirá siendo como lo ha sido Pedro Muguruza, la casa de todos vosotros y allí seguimos todos los mismos.
Tomad nota: ENVIAD TODAS LAS CONTRIBUCIONES, PROGRAMAS, MAGIA, CARTAS, ETC., ETC., a
BARQUILLO, 21 - 3º
28004 MADRID
Tfnos.: 231 23 85 ó 231 23 86

ADIOS Y BIENVENIDA

Le decimos adiós a José López —Pepe para nosotros y los amigos— que se marcha para establecerse por su cuenta.
Le deseamos toda la suerte del mundo, y damos la bienvenida al equipo, a Juan Márquez —"Juanito"—.

LAS AVENTURAS DE NUESTROS VIAJEROS
Joan y Diego están volviendo de Estados Unidos en este momento y Juan Mejuto se irá a Londres en un futuro no muy lejano. En el próximo número os contaremos sus aventuras y experiencias viajeras y commodorianas.

CONCURSOS PERMANENTES

Colaboraciones de Programas y Magia

La verdad sea dicha: os estáis portando; las colaboraciones que nos están llegando son numerosas y de muy buena calidad. Por lo tanto, vamos a aclarar las condiciones de nuestros concursos permanentes.

1) Los sorteos se realizarán en los meses de junio y diciembre.

2) Se sortearán siempre seis premios, tres de 15.000, 10.000 y 5.000 pesetas en metálico de "Commodore World" y tres del mismo valor en material de Microelectrónica y Control, entre todas las colaboraciones publicadas.

3) Se sortearán siempre seis premios, tres de 3.000, 2.000 y 1.000 pesetas en metálico de "Commodore World" y tres del mismo valor en material de Microelectrónica y Control, entre todas las contribuciones publicadas en la sección de "Magia".

4) Adicionalmente, se sortearán diversos premios de cintas, juegos, etc. donados por firmas y distribuciones, entre todas las colaboraciones.

5) Los autores de las colaboraciones vendidas dentro de nuestro "Servicio de Cintas" percibirán el 20% del precio de la cinta.

6) Todas las colaboraciones deben venir escritas a máquina y los programas grabados en cinta (si es posible), o con el listado completo en impresora. Nuestros lectores más jóvenes pueden escribir a mano pero con letra muy clara.

7) Quedan automáticamente descalificadas tanto del sorteo como del "Servicio de Cintas" las colaboraciones que hayan sido enviadas a otras revistas.

8) Las colaboraciones se enviarán a "Commodore World", *C/Barquillo, 21-3 izda. 28004 Madrid.*

MAS VIAJES Y OTRO REGALO DE "IMPRESION"... A POR EL SUSCRIPTOR 8.000

Es evidente que a nuestros sucriptores les gusta viajar. De momento Joan Pujiula visitará los Estados Unidos en julio y Juan Mejuto Iglesias irá a Inglaterra en el otoño. Queríamos sortear algo un poco especial ENTRE TODOS LOS SUCRIPTORES cuando lleguemos al suscriptor número 8.000, y hemos optado por dos premios, uno concedido por Microelectrónica y Control y otro concedido por "Commodore World".

MEC regalará una impresora MPS 801 (o similar), y nosotros hemos escogido un surtido de viajes muy "chulos" para dos personas para que el ganador elija el que más le apetezca. La selección que hemos hecho es la siguiente:

Viajes de 4 días en hoteles de primera clase a uno de estos lugares: París, Lisboa, Atenas, Amsterdam, Niza, Viena.
O viajes de una semana en hoteles de 4/5 estrellas: Canarias o Mallorca.

Convocatoria del Primer Concurso Nacional Escolar de Programación patrocinado por Commodore World y Microelectrónica y Control

BASES

1º Podrán participar en el mismo todos aquellos niños y jóvenes que estén en edad escolar pre-universitaria o equivalente.

2º Habrá tres grupos definidos:
a) E.G.B.
b) B.U.P.
c) F.P.

3º Los trabajos pueden ser presentados por equipos o por individuales.

4º Los trabajos presentados por equipos deben especificar:
a) Nombre del equipo.
b) Centro al que representan.
c) Nivel de enseñanza que cursan (E.G.B.-B.U.P-F.P.).
d) Nombre del profesor, monitor o coordinador.

5º Los trabajos presentados por individuales deben ir acompañados de:
a) Nombre del autor.
b) Edad y curso que estudia.
c) (¡) Si concursa representando a un centro: nombre del mismo y nombre del profesor responsable.
(¡¡) Si concursa por sí mismo: nombre del colegio donde estudia y firma del director o profesor responsable avalando que en tanto cuanto conoce al concursante el trabajo que presenta es suyo propio.

6º El tema del concurso es la realización de un programa educativo sobre materia libre a elegir por los concursantes. La extensión del programa es también de libre elección.

7º Cada centro puede presentar el número de equipos o concursantes individuales que desee, siempre que cada trabajo sea enviado por separado.

8º Los trabajos deben incluir:
a) Nombre el programa.
b) Nombre del computador para el que está confeccionado y periféricos adicionales que se necesitan (si fuese el caso).
c) Objetivo del programa.
d) Explicación detallada del programa (a máquina o impresora).
e) Listado completo a impresora.
f) Grabación en cinta o disco.

9º Todos los trabajos presentados deben ser originales e inéditos.

10º Los centros que pueden presentar equipos o concursantes individuales son:
a) Todo colegio, escuela o instituto, público o privado que imparta las enseñanzas de E.G.B., B.U.P, o F.P.
b) Centros de Informática que tengan grupos de enseñanza específica para niños y jóvenes menores de 18 años.
c) Clubs de informática que mantengan actividades específicas para niños y jóvenes menores de 18 años.

11º Los trabajos deben ser enviados antes del 31 de diciembre de 1984.

12º El fallo del concurso se dará a conocer en la primera semana de marzo de 1985 en el lugar, fecha y hora que se **indicarán**.

13º Los trabajos premiados quedarán propiedad de los patrocinadores.

Ver más abajo lugar, fecha y hora de la presentación oficial y anuncio de los premios.

Convocatoria del primer concurso nacional de programación en 80 columnas de Commodore-64 y 40-80 columnas del VIC-20 patrocinado por Ferre-Moret, S.A. Con la colaboración de Commodore-World, más de 500.000 ptas. en premios

BASES

1º Podrán participar todos los usuarios de COMMODORE-64 y VIC-20 que hayan obtenido la tarjeta de 80 columnas de FERRE-MORET, S.A.

2º Cada concursante podrá presentar, un programa realizado por él o por un equipo del que deberán indicar un titular.

3º Se enviará la explicación detallada del programa redactado en español, escrito a máquina. El programa se enviará en cassette o disco, adjuntando su correspondiente listado del programa.

4º Se enviarán por correo certificado a la siguiente dirección, FERRE-MORET, S.A. calle Buenos Aires nº 30, 2º-3.ª de BARCELONA-36. Antes del 31 de enero de 1985.

5º Los programas no premiados podrán ser solicitados por sus propietarios en el plazo de treinta días después del fallo.

6º El resultado se comunicará, directamente a los ganadores y se publicará en la revista COMMODORE WORLD.

7º Los premios se entregarán durante la primera semana de marzo de 1985 coincidiendo con la entrega de premios del PRIMER CONCURSO NACIONAL ESCOLAR DE PROGRAMACION.

8º Los programas premiados quedarán en propiedad de FERRE-MORET, S.A.

9º Los programas que no obtengan premio, recibirán, en caso que así lo considere FERRE-MORET, S.A., ofertas de compra de sus programas.

10º Los premios ascenderán a 500.000 Ptas. repartidos de la siguiente manera:

1º 250.000 Ptas.
2º 150.000 Ptas.
3º 100.000 Ptas.

11º Todos los participantes, que no hayan recibido premio u oferta de compra, podrán participar en el concurso de colaboradores que habitualmente COMMODORE WORLD viene realizando con un premio extra de FERRE-MORET, S.A., de 50.000 Ptas.

12º Los trabajos deberán de ser inéditos y desarrollar temas de utilidad, programas profesionales, científicos, educativos, etcétera.

FERRE-MORET, S.A. se reserva el derecho de modificar las bases de este certamen, dejarlo desierto o suprimirlo, por causa de fuerza mayor. En este caso la empresa se compromete a comunicarlo a través de COMMODORE WORLD.

La presentación oficial del 1er Congreso Nacional Escolar Escolar de Programación tendrá lugar en la sede de la FUNDACION.
ORTEGA Y GASSET - Calle Fortuny, 53, de Madrid, el día 20 de Septiembre, jueves a las 7,30 de la tarde.

Clave para interpretar los listados

Todos los listados que se publican en esta Revista han sido ejecutados en el modelo correspondiente de la gama de ordenadores COMMODORE. Para facilitar la edición de los mismos en la Revista y para mejorar su legibilidad por parte del usuario, se les ha sometido a ciertas modificaciones mediante un programa escrito especialmente para ello. Para los programas destinados a los ordenadores VIC-20 y COMMODORE 64, en los que se usan frecuentemente las posibilidades gráficas del teclado, se han sustituido los símbolos gráficos que aparecen normalmente en los listados por una serie de letras entre corchetes [] que indican la secuencia de teclas que se deben pulsar para obtener el carácter deseado. A continuación se da una tabla para aclarar la interpretación de las indicaciones entre corchetes:

[CRSRD] = Tecla cursor hacia abajo (sin SHIFT)

[CRSRU] = Tecla cursor hacia arriba con SHIFT)

[CRSRR] = Tecla cursor a la derecha (sin SHIFT)

[CRSRL] = Tecla cursor a la izquierda (con SHIFT)

[HOME] = Tecla CLR/HOME (sin SHIFT)

[CLR] = Tecla CLR/HOME (con SHIFT)

Las indicaciones [BLK] a [YEL] corresponden a la pulsación de las teclas de 1 a 8 junto a la tecla CTRL. Lo mismo sucede con [RVSON] y [RVSOF] respecto a la tecla CTRL y las teclas 9 y 10.

El resto de las indicaciones constan de la parte COMM o SHIF seguidas de una letra, número o símbolo —por ejemplo [COMM+] o [SHIFA]—. Esto indica que para obtener el gráfico necesario en el programa deben pulsarse simultáneamente las teclas COMMODORE (la que lleva el logotipo) o una de SHIFT y la tecla indicada por la letra, el número o el símbolo, en el ejemplo anterior: COMMODORE y + o SHIFT y A, respectivamente.

En los signos gráficos además se cuenta el número de veces que aparece. Por ejemplo, [7 CRSRR] equivale a 7 pulsaciones de la tecla cursor a la derecha y [3 SPC] tres pulsaciones de la barra espaciadora. ■



METEDURAS DE PATA



• **En el número 5, en el programa "MAT+DET", apareció varias veces el error entre corchetes "[ERROR]", todas las veces debía** aparecer la clave de las teclas Commodore y 7 "[COMM7].

• **En el número 6, en las líneas 180, 550, 560, 2000, 2020, 2030, y 2040, se cambió** [COMM4] POR [ERROR], y en la línea 590 debió ser [COMM7].

• **El programa "Tragaperras", indicaba que era para el C-64, debía decir VIC-20.**

• **El programa "Boggle", lo probamos con uno de los nuevos C-64 con la ROM versión 3, y funcionó correctamente, pero para que el cuadro de puntos y letras aparezca en la parte izquierda de la pantalla en los antiguos C-64, deben cambiarse las siguientes líneas:**

```
30 FOR T=1TO10:POKE A,46:POKE A+54272,1:A=A+1:NEXT T
70 FOR T=1TO10:POKE A,46:POKE A+54272,1:A=A+2:NEXT T
130 POKE A,G;POKE A+54272,1:A=A:Q(E)=G
```

• **El programa que apareció en la pág. 44 del nº 5 que no era lo suficiente legible en la impresión.** ▶

```
0 REM ****************           *VIC   M
EMORIA*        *(C)A.S.V.1984*
***************
10 DATA162,,160,32,24,32,156,255,169
,32,141,130,2,169,240,141,5,144,173,
2,144,9,128,141
11 DATA2,1 4,169,30,141,136,2,169,12
8,141,138,2,169,8,141,15,144,169,1,1
41,134,2,32,120
12 DATA227,96
20 FORI=10000TO10049:READA:POKEI,A:N
EXT:SYS10000
```

INDICE DE ANUNCIANTES

algunos de nuestros programas para los ordenadores personales

commodore









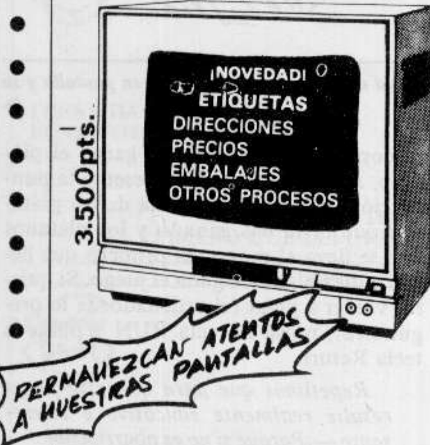
















B.M.

BASIC MICRO-ORDENADORES, S.A.

AVD. CESAR AUGUSTO, 72
Tlfs. 23 56 82 y 22 65 44
ZARAGOZA-3

APRENDE INGLES CON EL MONO MARCHOSO

Este programa ayuda a los "peques" —"papis" y "profes" también— a escribir en inglés. El uso obligatorio del diccionario os acostumbrará a su manejo, y, a la vez, lo pasaréis muy bien jugando con el mono.

El objetivo de este juego no es el simple recordatorio visual de un número determinado de palabras inglesas; su finalidad es el aprendizaje de tantas palabras inglesas como quieras usando el diccionario.

La utilización del diccionario es indispensable, tanto para las palabras que escribe el mono como para las que escribes tú, ya que el programa no sabe comprobar su validez.

Al mono se le puede engañar fácilmente —de hecho, siempre que se quiera— pero en venganza, tú no aprenderás ningún inglés y te aburrirás "como un mono", ganándole todo el tiempo porque sí.

¿El mono te está engañando,
o realmente sabe inglés?
Busca en tu diccionario,
y escribe las palabras que ves.

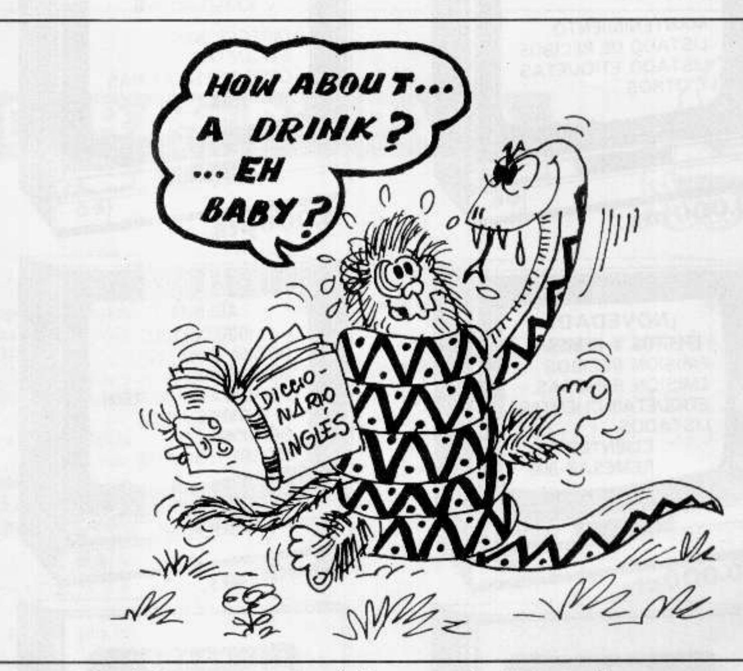
Después de cargar el programa, aparece en pantalla este "poema". En el tiempo que tardas en leerlo, el ordenador diseña los caracteres personalizados del juego y los almacena en memoria. La pantalla cambia, y empieza el juego.

La cara del mono aparece en la esquina superior izquierda de la pantalla. "Funky Monkey Spells..." (El Mono Marchoso Escribe...") aparece a la derecha del mono. Debajo de este mensaje aparecen tres caracteres subrayados en rojo.

El ordenador te pregunta si el mono escribió una palabra correcta (Did Funky spell a real word?) y te indica que pulses la tecla Y para SI y N para NO (Press Y or N) comprobando en tu diccionario antes de contestar.

Si el Mono Marchoso escribe una palabra que existe, él gana un plátano. Si el

VIC-20



A lo largo del artículo aparecen unas pantallas mostrando el texto real que aparece en pantalla y su traducción en español.

Mono no escribe una palabra correcta, y tu pulsas N, el ordenador te pregunta si puedes formar una palabra que contenga esas letras. Usa tu diccionario de inglés para encontrar una palabra correcta y forma una palabra cambiando el orden de las letras o añadiendo unos caracteres nuevos (la palabra puede ser de tres letras o ser tan larga como tú quieras), pulsando a continuación la tecla Y. El ordenador te indica que teclees la palabra y que pulses la tecla Return.

A continuación, el ordenador te pide que compruebes si la palabra es correcta o no —es decir, que contenga las tres primeras letras que te dio el mono— y que la palabra existe. Si te equivocaste o la palabra no existe, debes pulsar N y el Mono Marchoso gana el plátano. Pero si es correcta y pulsas Y, tú ganas el plátano. A continuación, se presenta la puntuación, que lleva la cuenta de los plátanos que tú te has ganado y los plátanos que se lleva el mono. El primero que llegue a diez plátanos gana el juego. Si quieres volver a jugar (el ordenador te lo preguntará), pulsa la tecla RUN y pulsa la tecla Return.

Repetimos que para que este juego resulte realmente educativo e interesante —¡Porque si no es aburridísimo!— exige el uso de un diccionario. Aprovechando el interés del niño, sería buena idea que este apuntara las palabras nuevas encontradas en el diccionario y que buscara la traducción de inglés al español. Así que, adelante y "Good Luck!" ■

```
10 PRINT""
20 REM: FUNKY MONKEY
30 POKE36869,240
40 PRINTCHR$(147)CHR$(28)TAB(53)"XXX
X
50 PRINTCHR$(31)TAB(24)"EL[2SPC]MONO
[2SPC]TE[2SPC]ESTA[SPC][2SPC][CRSRD]
[SPC]ENGAN[CRSRU][CRSRL][SHIFTR][CRSRD]
ANDO?"
60 PRINTTAB(24)"O[2SPC]REALMENTE[2SPC]
SABE[4SPC][CRSRD]INGLES?"
70 PRINTTAB(24)"BUSCA[4SPC]EN[4SPC]T
U[5SPC][CRSRD]DICCIONARIO,"
80 PRINT"[CRSRD][2SPC]Y[3SPC]ESCRIBE
[3SPC]LAS[4SPC][CRSRD][SPC]PALABRAS[SPC]
QUE[SPC]VES."
90 PRINTCHR$(28)TAB(31)"XXXX
100 FORT=1TO2000:NEXT
110 POKE52,28:POKE56,28:CLR
120 FORI=7168TO7679:POKEI,PEEK(I+256
00):NEXT
130 READA:IFA<0THEN340
140 FORC=ATOA+7:READJ:POKEC,J:NEXT
150 GOTO130
160 DATA7512,0,0,0,0,0,1,10,22
170 DATA7528,0,31,32,64,128,0,0,1
180 DATA7392,0,128,64,32,32,32,64,12
8
190 DATA7168,18,18,18,19,11,7,3,15
200 DATA7504,31,127,255,255,221,8,0,
0
210 DATA7408,0,128,192,192,192,192,1
28,224
220 DATA7384,18,34,34,35,19,18,10,6
230 DATA7400,102,102,127,129,20,20,0
,0
240 DATA7656,144,144,136,136,144,80,
96,64
250 DATA7648,2,2,2,1,1,0,0,0
260 DATA7664,129,66,60,24,1,194,60,0

270 DATA7544,64,64,64,128,0,0,0,0
280 DATA7416,255,255,255,255,255,255
```



FUNKY MONKEY SPELLS ...
EL MONO MARCHOSO ESCRIBE...

FUNKY HAS
FUNKY TIENE

YOU HAVE
TU TIENES

1ST SCORE OF 10 WINS
EL PRIMERO QUE LLEGUE A DIEZ, GANA.



DID FUNKY SPELL A REAL WORD?
¿FUNKY ESCRIBIO UNA PALABRA CORRECTA?

PRESS Y or N - PULSA Y o N



THEN FUNKY GETS A BANANA!
¡FUNKY GANA UN PLATANO!

YOU GET A BANANA!
¡TU GANAS UN PLATANO!



CAN YOU MAKE A REAL WORD OUT OF FUNKY'S

¿TU PUEDES FORMAR UNA PALABRA?

```
,255,255
290 DATA7488,255,255,0,0,0,0,0,0
300 DATA7448,32,112,63,31,31,15,15,7

310 DATA7456,0,0,255,255,255,255,255
,255
320 DATA7464,0,0,252,254,254,252,248
,240
330 DATA-1
340 P=1000:FS=0:YS=0:POKE36869,255
350 PRINTCHR$(147)CHR$(28)"←←←←←←←←"
360 FORT=1TO6:PRINT"←"SPC(5)"←":NEXT
370 PRINT"←←←←←←←←"
380 C$="+-£[CRSRD][3CRSRL]@*↑[CRSRD]
[3CRSRL][]=[CRSRD][3CRSRL]<>/"
390 PRINTCHR$(19)CHR$(144)TAB(46)C$
400 PRINTCHR$(19)TAB(52)"FUNKY[SPC]M
ONKEY"
410 FORT=1TOP:NEXT
420 PRINTTAB(9)"SPELLS...":FORT=1TOP
:NEXT
430 GOSUB660
440 PRINTCHR$(19)CHR$(31)TAB(122)CHR
$(C);
450 GOSUB660
460 IFC=65ORC=69ORC=73ORC=79ORC=85TH
EN480
470 GOTO450
480 PRINTCHR$(19)TAB(123)CHR$(C);
490 GOSUB660
500 PRINTCHR$(19)TAB(124)CHR$(C)
510 PRINTCHR$(19)CHR$(28)TAB(144)"((
("
520 PRINTCHR$(19)CHR$(144)TAB(255)TA
B(155);
530 PRINT"FUNKY[SPC]HAS[3SPC]YOU[SPC]
HAVE"
540 PRINTCHR$(17)FSCHR$(158)"#$%"CHR
$(144)"'S";
550 PRINTSPC(4)YSCHR$(158)"#$%"CHR$(
144)"'S
```

```
560 PRINTCHR$(17)CHR$(144)CHR$(29)"1
ST[SPC]SCORE[SPC]OF[SPC]10[SPC]WINS"
;
570 PRINTCHR$(145)
580 PRINTCHR$(19)CHR$(31)TAB(244);
590 PRINT"DID[SPC]FUNKY[SPC]SPELL[SPC]
A[5SPC]REAL[SPC]WORD?"
600 PRINTTAB(24)CHR$(30)"PRESS[SPC]Y
[SPC]OR[SPC]N"
610 GETG$:IFG$=""THEN610
620 IFG$="N"THEN680
630 IFG$="Y"THEN740:GOTO610
640 GETG$:IFG$=""THEN640
650 GOTO430
660 C=INT(RND(.)*26)+65:RETURN
670 FORT=1TOP:NEXT:RETURN
680 GOSUB870
690 PRINTCHR$(19)CHR$(31)TAB(243);
700 PRINT"CAN[SPC]YOU[SPC]MAKE[SPC]A
[SPC]REAL[3SPC]WORD[SPC]OUT[SPC]OF[SPC]
FUNKY'S?"
710 GETG$:IFG$=""THEN710
720 IFG$<>"Y"ANDG$<>"N"THEN710
730 IFG$="Y"THEN880
740 GOSUB870
750 FS=FS+1:PRINTTAB(198)CHR$(144)FS
760 IFFS=10THEN1070
770 PRINTCHR$(19)CHR$(31)TAB(244);
780 PRINT"THEN[SPC]FUNKY[SPC]GETS[SPC]
A[5SPC]BANANA!"
790 P=2000:GOSUB670
800 GOSUB870
810 PRINTCHR$(19)CHR$(31)TAB(244);
820 PRINT"ARE[SPC]YOU[SPC]READY[SPC]
FOR[5SPC]ANOTHER[SPC]WORD?"
830 GETG$:IFG$=""THEN830
840 IFG$<>"N"ANDG$<>"Y"THEN830
850 IFG$="N"THEN830
860 IFG$="Y"THENP=500:GOTO350
870 PRINTCHR$(19)TAB(242);:FORI=1TO4
4:PRINT"[SPC]";:NEXT:RETURN
880 GOSUB870
```

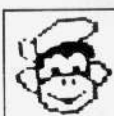


TYPE WORD!
PRESS RETURN

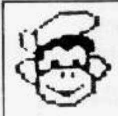
TECLEA LA PALABRA
PULSA RETURN



IS... RIGHT?
¿ES... CORRECTA?



ARE YOU READY FOR ANOTHER WORD?
¿QUIERES OTRA PALABRA?



FUNKY WINS!
¡FUNKY GANA!

YOU WIN!
¡TU GANAS!

```
890 PRINTCHR$(19)CHR$(30)TAB(248)TAB
(68)"PRESS[SPC]RETURN"
900 PRINTCHR$(19)CHR$(31)TAB(244)
910 INPUT"TYPE[SPC]WORD";A$
920 GOSUB870
930 PRINTCHR$(19)TAB(244)TAB(68)CHR$
(30)"PRESS[SPC]Y[SPC]OR[SPC]N"
940 PRINTCHR$(19)CHR$(31)TAB(244)"IS
[SPC]"A$"[SPC]RIGHT?"
950 GETG$:IFG$=""THEN950
960 IFG$<>"Y"ANDG$<>"N"THEN950
970 IFG$="N"THEN740
980 GOSUB870
990 YS=YS+1:PRINTTAB(210)CHR$(144)YS

1000 GOSUB870
1010 IFYS=10THEN1050
1020 PRINTCHR$(19)CHR$(31)TAB(244)"Y
OU[SPC]GET[SPC]A[SPC]BANANA!"
1030 P=2000:GOSUB670
1040 GOTO800
1050 PRINTCHR$(19)CHR$(31)TAB(248)"Y
OU[SPC]WIN!"
1060 GOTO1100
1070 GOSUB870
1080 PRINTCHR$(19)CHR$(31)TAB(246)"F
UNKY[SPC]WINS!"
1090 GOTO1100
1100 PRINTCHR$(144):FORJ=1TO12
1110 PRINTCHR$(19)TAB(46)"[3SPC]"TAB
(24)"[3SPC]"TAB(24)"[3SPC]";
1120 PRINTTAB(24)"[3SPC]"TAB(24)"[3SPC]
"
1130 PRINTCHR$(19)TAB(24)C$:FORT=1TO
150:NEXT
1140 PRINTCHR$(19)TAB(24)"[3SPC]"TAB
(24)"[3SPC]";
1150 PRINTCHR$(19)TAB(24)"[3SPC]"TAB
(24)"[3SPC]"
1160 PRINTCHR$(19)TAB(46)C$:FORT=1TO
150:NEXT:NEXT
1170 GOTO1170
```

casa de software s.a.

PARA COMMODORE 64

PRACTICALC 64



MANUAL EN CASTELLANO

PROCESADOR DE TEXTO

35.000 caracteres, 240 columnas
Versión diskette: 21.500,–
Versión cartucho: 24.900,–
(grabación de documentos en diskette y cassette)

HOJA DE CALCULO

2.000 coordenadas
(funciones matemáticas, sort, gráficos...)
Versión cassette C-64: 15.500,–
Versión diskette C-64: 17.500,–
Versión cassette VIC-20: 11.500,–
Versión diskette VIC-20: 13.500,–

VIZAWRITE 64

MANUAL EN CASTELLANO

CONTABILIDAD 64 PROFESIONAL



VERSION: A	VERSION: B
300 cuentas	600 cuentas
3.000 apuntes	2.300 apuntes

- Listado de diario
- Balance de situación y de Sumas y Saldos
- Estractos de cuenta
- Cuenta de explotación, diario de cierre
- Mantenimiento de ficheros, utilitario...

EQUIPO NECESARIO:

- COMMODORE 64 y unidad de disco
- IMPRESORA MPS 801 (versión A y B)
- IMPRESORAS CENTRONICS (versión B)

P.V.P.: VERSION A: 24.550,–
VERSION B: 29.500,–

ADAPTADO AL PLAN GENERAL CONTABLE ESPAÑOL

PROGRAMAS EN CARTUCHO + DISKETTE

VIDEOCLUBS 64

1.800 videofilms
650 clientes por disco
Saldo de cada cliente
Altas y bajas videofilms
Búsqueda videofilms/clientes.

P.V.P.: 35.000,–

PROGRAMAS EN CASSETTE

CONTABILIDAD PERSONAL	6.000,–
CONTABILIDAD DOMESTICA	3.500,–
CONTABILIDAD DOMESTICA (VIC-20)	2.500,–
PERT	4.000,–
CALCULOS FINANCIEROS	3.000,–
CALCULOS ESTADISTICOS	1.900,–
FICHERO MONEDAS	3.000,–
FICHERO SELLOS	3.000,–
FICHERO BIBLIOTECA	3.000,–
FICHERO AGENDA	3.000,–
QUINIELAS	3.000,–

ADMINISTRACION DE FINCAS

30 inmuebles
400 inquilinos
Listado recibos
Propiedad vertical
Comunidades de propietarios

P.V.P.: 35.000,–

LAPIZ OPTICO Y SOFTWARE GRAFICO

P.V.P.: 12.500,–

JOY STICK DE PRECISION

P.V.P. 2.950,–

DE VENTA EN TIENDAS Y DISTRIBUIDORES ESPECIALIZADOS

DESEO RECIBIR INFORMACION:

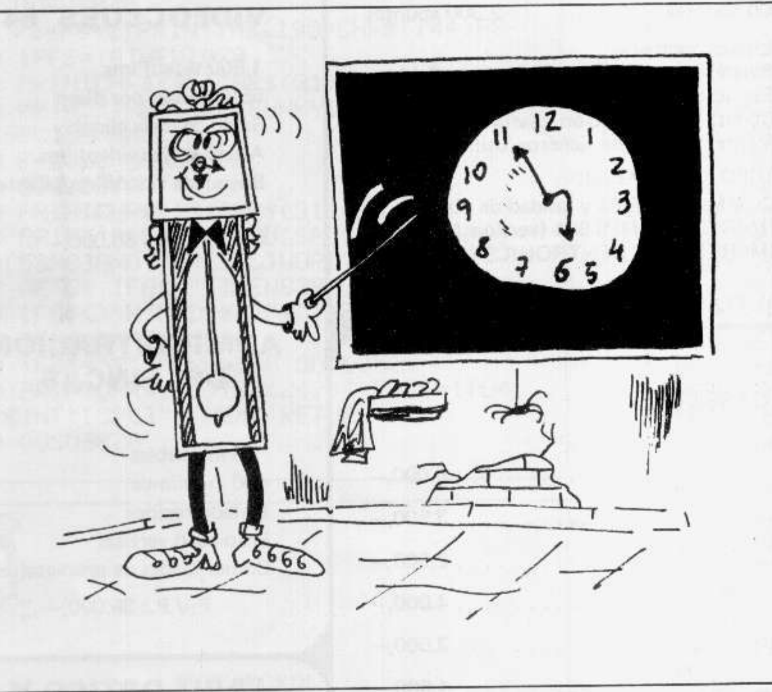
Nombre y dirección:

casa de software, s.a.
c/. aragón, 272, 8.°, 6.ª
tel. 215 69 52
08007 barcelona

COMODORE 64 ES UNA MARCA REG. DE COMMODORE BUSSINES MACHINES, INC.

ES LA "HORA" DE APRENDER

Este programa sirve para enseñar la hora a los niños. El objeto de este juego educativo es de parar las manos del reloj para que coincidan con la hora indicada en pantalla.



VIC-20
NO AMPLIADO
C-64

Profesores y padres pueden utilizar este programa para enseñar la hora a los niños. Se presenta en forma de juego, el niño gana si logra parar las manillas del reloj tres veces en la hora indicada. El programa fue escrito para el VIC no ampliado; también se incluye la versión para el C-64.

El programa empieza con un diagrama de un reloj que muestra cómo los números de la hora corresponden a intervalos de cinco minutos. Este diagrama se presenta en pantalla si el niño falla tres veces consecutivas.

Además, esta pantalla le pregunta al niño si necesita instrucciones. Si no son necesarias, se generan unas horas al azar bajo las mismas condiciones de dificultad elegidas al principio. Si el niño prescinde de instrucciones y nivel de dificultad, se elige automáticamente el nivel más sencillo.

Las instrucciones son muy sencillas. Se trata de intentar parar el reloj cuando las manillas coincidan con la hora en pantalla. El jugador gana cuando acierta tres veces. Las manillas del reloj se mueven automáticamente. Pueden ser avanzadas mediante el uso de dos de las teclas de funciones: la tecla F1 hace que la manilla que indica la hora se desplace más de prisa, y la tecla F3 que se desplace la manilla de los minutos más de prisa. Esto significa que el niño puede colocar las manillas del reloj en una posición aproximada sin tener que esperar.

La mejor forma de utilizar las teclas de funciones es pulsarlas después de cada "tick-tock" para evitar cualquier salto inesperado.

También ayuda colocar la manilla que indica la hora justo antes de la hora anunciada. Así se puede utilizar la manilla de los minutos para acercarse aún más.

Es aconsejable dejar que el reloj se coloque automáticamente en la posición final. Se presenta la tecla F7 como la que se utiliza para iniciar y parar el movimiento del reloj pero, de hecho, la tecla F5 o cualquier otra tecla normal funciona igual de bien.

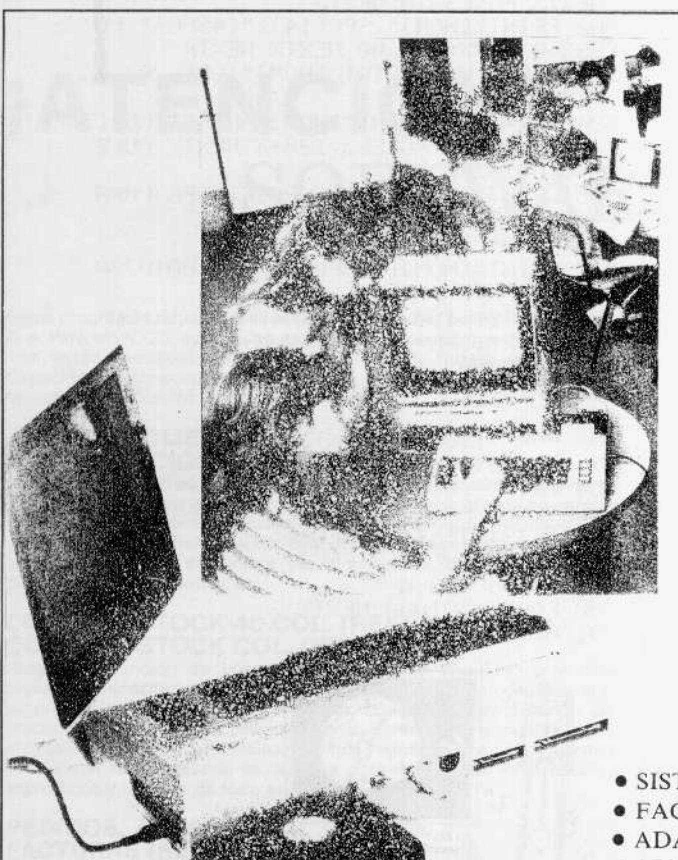
Cada vez que se presenta el reloj en pantalla, una hora generada al azar aparece en la esquina superior. Hay tres niveles de dificultad. El nivel uno solamente genera las horas en punto (1:00, 2:00, 3:00, etc.). Las medias (4:30, 5:30, 6:30, etc.) aparecen en el nivel dos. El nivel tres presenta los múltiples de cinco minutos después de la hora (7:35, 8:15, 9:10, etc.).

Cuando se para el reloj, el programa calcula las posiciones de las manillas y la hora indicada. Si la relación es correcta, suena la alarma y aparece la palabra "Sí" junto con el número de intentos correctos hasta ese momento. Cuando el niño acierta tres veces, suena una sirena, y el jugador puede elegir otro nivel o dejar de jugar.

La versión para el C-64 es un poco más larga, pero el formato del programa es idéntico (las funciones de pantalla y sonido requieren más Pokes).

Espero que este programa resulte tan educativo como divertido. ■

```
0 REM PARA EL RELOJ VIC-20
1 L$="7826784979158003802480208017799
947906784078197822"
2 LH$="782778717937802580238020F0167
97279067818782O7822"
3 IH$="+2178+2178-0164-2377-2377-226
6-2178-2178+0164+2377+2377+2266"
4 IP$="+2178+2178-0164-2377-2377-226
6-2178-2178+0167+2377+2377+2266"
5 B$="77787784780779178049807080648O
5980367904779477 3"
6 CO=30720:V=36878:S1=36874:S2=36875
:S3=36876:SC=36879:POKEV,15
10 POKESC,174:PRINT"[CLR][3CRSRR][RVSON]
DIAGRAMA[SPC]DE[SPC]RELOJ[RVSOFF]":P
RINT"[CRSRD]";SPC(8);"00":PRINT"[6SPC]
[5COMM@]5"
15 PRINT"[3SPC]55[SHIFTN][2SPC]12[SPC]
[SHIFTM]":PRINT"[4SPC][SHIFTN]11[4SPC]
1[SHIFTM][SPC]10":PRINT"[SPC]50[SHIFTN]
[9SPC][SHIFTM]"
20 PRINT"[2SPC][COMMM]10[SPC]RELOJ[2SPC]
2[COMMG]":PRINT"[2SPC][COMMM]";SPC(1
1);"[COMMG]"
25 PRINT"45[COMMM]9[2SPC][SHIFTQ][SHIFT*]
[COMM+][5SPC][COMMG]15":PRINT"[2SPC]
[COMMM][5SPC][SHIFT-][5SPC][COMMG]"
30 PRINT"[2SPC][COMMM]8[4SPC][SHIFT-]
[4SPC]4[COMMG]":PRINT"[SPC]40[SHIFTM]
[4SPC][SHIFT-][4SPC][SHIFTN]20"
35 PRINT"[4SPC][SHIFTM]7[2SPC][SHIFTQ]
[2SPC]5[SHIFTN]":PRINT"[3SPC]35[SHIFTM]
[2SPC]6[2SPC][SHIFTN]25"
40 PRINT"[6SPC][5COMMT]":PRINTTAB(7)
"30"
45 PRINT"[CRSRD][4SPC]HORA=[SPC]9:30
"
50 PRINT"[CRSRD]PULSA[SPC]'F7'PARA[SPC]
EMPEZAR[CRSRD]O[SPC]'I',[SPC]INSTRUC
CIONES.";
55 GETS$:IFS$=""THEN55
56 IFS$="I"THEN58
57 GOTO200
58 PRINT"[CLR][4SPC][RVSON][RED]INST
RUCCIONES[RVSOFF][BLU]":PRINT"[CRSRD]
INTENTA[SPC]PARAR[SPC]EL[SPC]RELOJ"
65 PRINT"CUANDO[SPC]LAS[SPC]MANILLAS
[SPC]":PRINT"[CRSRD]INDIQUEN[SPC]LA[SPC]
HORA."
70 PRINT"[CRSRD]GANAS[SPC]CON[SPC]3[SPC]
ACIERTOS."
75 PRINT"[2CRSRD][RVSON]TECLAS[SPC]D
E[SPC]CONTROL[RVSOFF]":PRINT"[CRSRD]
'F1'[SPC]MUEVE[SPC]LA[SPC]MANILLA[6SPC]
CHICA"
80 PRINT"[CRSRD]'F3'[SPC]MUEVE[SPC]L
A[SPC]GRANDE"
85 PRINT"[CRSRD]'F7'[SPC]PARA[SPC]EL
[SPC]RELOJ"
87 PRINT"[2CRSRD](PULSA[SPC]UNA[SPC]
TECLA.)";
90 GETS$:IFS$=""THEN90
```



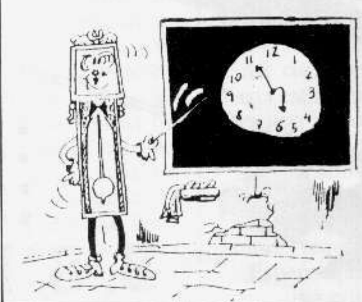


```
150 PRINT"[CLR][CRSRD][SPC][RVSON]DI
FICULTAD[SPC]DEL[SPC]JUEGO"
155 PRINT"[2CRSRD](1)[SPC]HORAS":PRI
NT"[CRSRD](2)[SPC]HORAS[SPC]Y[SPC]1/
2[SPC]HORAS"
160 PRINT"[CRSRD](3)[SPC]INTERVALOS[2SPC]
DE[2SPC]5[5SPC][CRSRD]MINUTOS."
165 PRINT"[5CRSRD]PULSA[SPC]1,2,O[SPC]
3.":PRINT"[2CRSRD]PULSA[SPC]'F'[SPC]
PARA[SPC]FIN."
170 GETD$:IFD$=""THEN170
173 IFD$="F"THENPOKEV,0:POKESC,27:PR
INT"[CLR]":END
175 D=VAL(D$)
180 IFD>3ORD<1THEN170
200 PRINT"[CLR]":R=0:W=0:POKESC,159
210 PRINT"[4CRSRR][14COMM@]"
211 PRINT"[2CRSRR][RVSON][SHIFT£][RVSOFF]
[SHIFT£]11[4CRSRR]12[5CRSRR]1[SHIFTM]
"
212 PRINT"[CRSRR][RVSON][SHIFT£][RVSOFF]
[SHIFT£]";SPC(16);"[SHIFTM]"
213 PRINT"[CRSRR][RVSON][SPC][RVSOFF]
10";SPC(15);"2[COMMG]"
214 PRINT"[CRSRR][RVSON][SPC][RVSOFF]
";SPC(18);"[COMMG]":PRINT"[CRSRR][RVSON]
[SPC][RVSOFF]";SPC(18);"[COMMG]":PRI
NT"[CRSRR][RVSON][SPC][RVSOFF]"SPC(1
8);"[COMMG]"
215 PRINT"[CRSRR][RVSON][SPC][RVSOFF]
";SPC(18);"[COMMG]":PRINT"[CRSRR][RVSON]
[SPC][RVSOFF]"SPC(18);"[COMMG]"
216 PRINT"[CRSRR][RVSON][SPC][RVSOFF]
9[7CRSRR][RVSON][2SPC][RVSOFF][7CRSRR]
3[COMMG]"
217 PRINT"[CRSRR][RVSON][SPC][RVSOFF]
[8CRSRR][RVSON][2SPC][RVSOFF][8CRSRR]
[COMMG]"
218 PRINT"[CRSRR][RVSON][SPC][RVSOFF]
";SPC(18);"[COMMG]":PRINT"[CRSRR][RVSON]
[SPC][RVSOFF]";SPC(18);"[COMMG]":PRI
NT"[CRSRR][RVSON][SPC][RVSOFF]";SPC(
18);"[COMMG]"
219 PRINT"[CRSRR][RVSON][SPC][RVSOFF]
";SPC(18);"[COMMG]":PRINT"[CRSRR][RVSON]
[SPC][RVSOFF]";SPC(18);"[COMMG]"
220 PRINT"[CRSRR][RVSON][SPC][RVSOFF]
8";SPC(16);"4[COMMG]":PRINT"[CRSRR][RVSON]
[SPC][RVSOFF]";SPC(18);"[COMMG]"
221 PRINT"[CRSRR][COMM*][RVSON][COMM*]
[RVSOFF]7[6CRSRR]6[6CRSRR]5[CRSRR][SHIFTN]
": PRINT"[3CRSRR][16COMMT]"
240 HR=INT(RND(1)*12+1):Q=0
242 IFD=3THENMN=INT(RND(1)*12):GOTO2
48
243 IFD=2THENMN=6*INT(RND(1)*2):GOTO
248
247 MN=0
248 IFMN=0ORMN=1THENPRINT"[HOM]";HR;
":[SPC]";RIGHT$("0"+RIGHT$(STR$(5*MN
),1),2):GOTO250
249 PRINT"[HOM]";HR;":";5*MN
250 H=1:RM=0
251 IFRM>2THENPL=VAL(MID$(LH$,(H-1)*
4+1,4)):GOTO280
270 PL=VAL(MID$(L$,(H-1)*4+1,4))
280 PB=VAL(MID$(B$,4*RM+1,4))
284 IFRM=0THENII=22:PP=66:GOTO290
285 II=VAL(MID$(IP$,(RM-1)*5+1,3))
286 PP=VAL(MID$(IP$,(RM-1)*5+4,2))
290 POKEPB,160:POKEPB+CO,0
300 FORK=1TO5:POKEPB+K*II,PP:POKEPB+
K*II+CO,0:NEXTK
305 POKES1,240:FORK=1TO10:NEXTK:POKE
S1,0
310 IFRM>2THENI=VAL(MID$(IH$,(H-1)*5
+1,3)):P=VAL(MID$(IH$,(H-1)*5+4,2)):
GOTO330
320 I=VAL(MID$(IP$,(H-1)*5+1,3)):P=V
AL(MID$(IP$,(H-1)*5+4,2))
330 POKEPL,160:POKEPL+CO,6
340 FORK=1TO3:POKEPL+K*I,P:POKEPL+K*
I+CO,6:NEXTK
350 FORK=1TO600:NEXTK
353 IFQ2=1THEN650
355 GETS$:IFS$=""THEN379
357 IFS$="[F1]"THENRM=RM+12:GOTO379
358 IFS$="[F3]"THENRM=RM+2:GOTO379
359 IFHR=HANDMN=RMTHEN500
360 GOTO600
379 POKES2,240:FORK=1TO10:NEXTK:POKE
S2,0
380 POKEPB,32:FORK=1TO5:POKEPB+K*II,
32:NEXTK
381 POKEPL,32:FORK=1TO3:POKEPL+K*I,3
2:NEXTK
385 IFQ=1THEN240
386 IFQ2=1THEN251
390 RM=RM+1
391 IFRM>11THENH=H+1:RM=0
392 IFH>12THEN250
400 GOTO251
500 FORK=1TO2:PRINT"[HOM]"SPC(14);"S
I[4SPC]";R+1;
501 FORJ=1TO15:FORX=1TO10:POKES3,230
:NEXTX:POKES3,0:NEXTJ
502 PRINT"[HOM]";SPC(14);"[4SPC]":PO
KES3,0:FORX=1TO200:NEXTX:NEXTK
503 R=R+1:W=0:PRINT"[HOM]";SPC(20);R

504 IFR=3THENPRINT"[HOM]FANTASTICO![7SPC]
":FORX=1TO50:POKES3,190+X:NEXTX:POKE
S3,0:GOTO700
505 PRINT"[HOM]PULSA[SPC]F7[SPC]PARA
[SPC]EMPEZAR.";
506 GETS$:IFS$=""THEN50?
510 PRINT"[HOM][22SPC]";:Q=1:GOTO380

600 POKES1,190:FORX=1TO400:NEXTX:POK
ES1,0:Q2=1:RM=MN:H=HR:W=W+1
603 GOTO380
650 PRINT"[HOM]";SPC(9);"CORREGIDO"
Q2=0
652 FORX=1TO2000:NEXTX
65? IFW>2THEN10
654 PRINT"[HOM]PULSA[SPC]F7[SPC]PARA
[SPC]EMPEZAR.";
655 GETS$:IFS$=""THEN655
?57 PRINT"[HOM][22SPC]";
660 Q=1:GOTO380
700 FORX=1TO1000:NEXTX
701 GOTO150
```

FIN



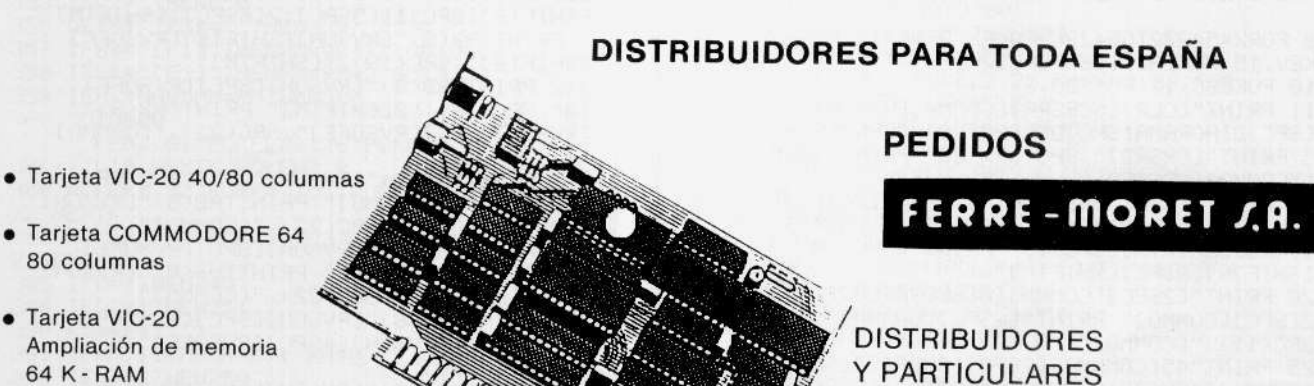
¡INCREIBLES TARJETAS PARA COMMODORES 64 Y VIC-20 DE 80 COLUMNAS!

LA TARJETA DE 64K ES LA PERFECTA COMBINACION PARA LA TARJETA DE 40/80 COLUMNAS:

- Puede escoger entre 22/40/80 columnas y obtiene 31743 bytes libres!!!

- El sofware del EPROM de la tarjeta le permite utilizar todos los comandos de ficheros (OPEN, CLOSE, PRINT, INPUT, GET, SAVE and LOAD) con los 31232 Bytes de Memoria «oculta», lo que produce la sensación de qué hay 8 cassettes super rápidos conectados a su Vic-20 Imagínese... menos de 1 segundo para guardar 28 K!!!

DISTRIBUIDORES PARA TODA ESPAÑA



- Tarjeta VIC-20 40/80 columnas
- Tarjeta COMMODORE 64 80 columnas
- Tarjeta VIC-20 Ampliación de memoria 64 K - RAM
- Slots de expansión para VIC-20 y 64 de 2 y 5 para conexión de tarjetas

PEDIDOS

FERRE-MORET S.A.

DISTRIBUIDORES Y PARTICULARES

Tel. 93/ 250 84 40/ Contestador Aut.
Escribir indicando la sección
C/. Buenos Aires n.º 30 2º 3ª
BARCELONA-36

¡ATENCION!
SOFTWARE 80 COLUMNAS

Estos programas requieren las tarjetas distribuidas por FERRE-MORET, S. A. Para el VIC-20, aunque los programas funcionan con un mínimo de 16K, están diseñados para su utilización con la Tarjeta de 64K RAM. Capacidad de almacenaje ilimitada. Todos los programas se venden con un manual explicativo en castellano.

AGENDA DOMESTICA Y DE NEGOCIOS (REF. ADD4080C)

Incluye una función de búsqueda que identifica la totalidad o parte de los nombres, direcciones o números de teléfono, es decir, el sistema permite identificar un nombre, dirección o teléfono, aunque únicamente se recuerde parte de la información. Si el usuario no recuerda una dirección concreta, llamando el nombre de la población, obtendrá todas las direcciones de esa población.

CONTROL STOCK 40 COL. (REF. st40C) CONTROL STOCK COL. (REF. st80C)

Posee una función de búsqueda que permite visualizar productos concretos, especificando el producto, su descripción, lugar de almacenaje, proveedor, dirección del proveedor, código de identificación del producto, precio de coste, precio de venta, cantidad (en paquetes, cajas, etc.), valoración del stock, niveles máximos y mínimos de stock. Si además se dispone de impresora, se pueden obtener listados de productos específicos y listados de todo el almacén.

PEDIDOS, ALBARANES, FACTURAS (REF. IDO4080CP)

Se precisa la tarjeta de 64K RAM impresora. El programa permite imprimir encabezamiento con el nombre de la Empresa y su dirección y se puede elegir que el documento sea una Factura, Albarán o Pedido. Otros detalles del encabezamiento son: número de factura, fecha, número de pedido del cliente, referencia del cliente, etc. El programa permite la introducción de varios ITE's, especifica el producto, descripción, cantidad, precio antes de impuestos, precios con impuestos incluidos, gastos de envío etc. Numeración automática de los documentos. Permite hacer copias.

DIETARIO DOMESTICO Y DE NEGOCIOS (REF. HBD80CDP)

Funciona con Cassette y Unidad de Disco. El programa pide la fecha e identifica cualquier entrevista concertada para ese día, detallando con quién es la entrevista, motivo de la misma, hora y lugar de la reunión, duración de la reunión, etc.
Permite la búsqueda diaria o mensual para identificar reuniones por la fecha, o lugar de reunión, o persona, tanto en el pasado como en el futuro, y además visualiza un calendario.

CONTROL DE ALMACEN A 80 COL. (REF. ST80CDP)

Los datos se pueden almacenar en cassette o Unidad de Disco. Se incluye la descripción de los productos, su referencia o número, cantidad, precio de compra, precio de venta, niveles máximo y mínimo de existencias, pedidos mínimos exigidos por los proveedores, nombre y dirección del proveedor, etc. Se dispone de una función de búsqueda, que permite visualizar todo el almacén, productos cuya cantidad exceda la máxima autorizada, productos con existencia por debajo del nivel mínimo requerido, salidas de almacén, valor del stock, precios y márgenes.

FICHERO DE PERSONAL A 80 COL. (REF. PRF80CDP)

Se necesita impresora. El programa incluye el nombre, dirección, teléfono, edad, fecha de nacimiento, número, sexo, estado, número de Seguridad Social, Cualificaciones, Historial de empleo, Estudios, sueldo, funciones que realiza actualmente, programas de training a los que ha asistido, etc. Se pueden visualizar todos los datos o seleccionar los que se desean, mediante la utilización de una función de búsqueda. Excelente para selección de personal, ya que permite identificación por nombre, pero también se puede buscar por una cualificación o característica concreta, por la edad, experiencia, etc.

```
0 REM PARA EL RELOJ-C64
1 L$="13271409152916891767176317601б
78151813981320132З"
2 LH$="13291449152917691762176317581
6381518131813201323"
3 IH$="+3978+3978-0145-4177-4177-406
6-3978-3978+0145+4177+4177+4066"
4 IP$="+3978+3978-0164-4177-4177-406
6-3978-3978+0167+4177+4177+4066"
5 B$="124312491331153117711849184318
381756151613161238":S2=54285
6 CO=54272:V=54296:SC=53281:BO=53280
:W2=54283:A2=54284:H2=54280:L2=54279

8 FORX=54272TO54296:POKEX,0:NEXTX:PO
KEV,15:POKEA2,116:POKES2,128
10 POKESC,12:POKEBO,3
11 PRINT"[CLR][5CRSRR][COMM4][RVSON]
[SPC]DIAGRAMA[SPC]DE[SPC]HORA[SPC][RVSOFF]
":PRINT"[CRSRD]";SPC(3);"00":PRINT"[6SPC]
[5COMM@][SPC]5"
15 PRINT"[3SPC]55[SHIFTN][2SPC]12[SPC]
[SHIFTM]":PRINT"[4SPC][SHIFTN]11[3SPC]
1[SPC][SHIFTM][SPC]19":PRINT"[SPC]50
[SHIFTN][9SPC][SHIFTM]"
20 PRINT"[2SPC][COMMM]10[SPC]RELOJ[SPC]
2[SPC][COMMG]":PRINT"[2SPC][COMMM]";
SPC(11);"[COMMG]"
25 PRINT"45[COMMM]9[2SPC][SHIFTQ][SHIFT*]
[COMM+][4SPC]3[COMMG]15":PRINT"[2SPC]
[COMMM][5SPC][SHIFT-][5SPC][COMMG]"
30 PRINT"[2SPC][COMMM]8[4SPC][SHIFT-]
[3SPC]4[SPC][COMMG]":PRINT"[SPC]40[SHIFTM]
[4SPC][SHIFT-][4SPC][SHIFTN]20"
35 PRINT"[4SPC][SHIFTM][SPC]7[SPC][SHIFTQ]
[SPC]5[SPC][SHIFTN]":PRINT"[3SPC]35[SHIFTM]
[2SPC]6[2SPC][SHIFTN]25"
40 PRINT"[6SPC][5COMMT]":PRINTTAB(7)
"30"
45 PRINT"[CRSRD][4SPC]HORA=[SPC]9:30
"
50 PRINT"[2CRSRD][SPC]PULSA[SPC]'F7'
[SPC]PARA[SPC]COMENZAR[SPC],O[2SPC]P
ULSA":PRINT"[SPC]'I'[SPC]PARA[SPC]IN
STRUCCIONES."
55 GETS$:IFS$=""THEN55
56 IFS$="I"THEN58
57 GOTO200
58 PRINT"[CLR]";SPC(11);"[RVSON][SPC]
INSTRUCCIONES[SPC][RVSOFF]"
70 PRINT"[CRSRD][SPC]GANAS[SPC]ACERT
ANDO[SPC]3[SPC]VECES.
75 PRINT"[2CRSRD][RVSON][SPC]TECLAS[SPC]
DE[SPC]CONTROL[SPC][RVSOFF]":PRINT"[2CRSRD]
'F1'[SPC]MUEVE[SPC]LA[SPC]MANILLA[SPC]
CHICA."
80 PRINT"[CRSRD]'F3'[SPC]MUEVE[SPC]L
A[SPC]MANILLA[SPC]GRANDE."
85 PRINT"[CRSRD]'F7'[SPC]PARA[SPC]EL
[SPC]RELOJ."
87 PRINT"[3CRSRD][SPC](PULSA[SPC]UNA
[SPC]TECLA.)"
90 GETS$:IFS$=""THEN90
150 POKESC,11:POKEBO,7:PRINT"[CLR][WHT]
[CRSRD][RVSON]DIFICULTAD[SPC]DE[SPC]
LA[SPC]PRUEBA[RVSOFF]"
155 PRINT"[CRSRD](1)[SPC]HORAS":PRIN
T"[CRSRD](2)[SPC]HORAS[SPC]Y[SPC]1/2
-HORAS"
160 PRINT"[CRSRD](3)[SPC]INTERVALOS-
5[SPC]MIN."
165 PRINT"[6CRSRD][2SPC]PULSA[SPC]1,
2,O[2SPC]3":PRINT"[CRSRD][SPC]O[2CRSRD]
[2SPC]PULSA[SPC]'F'[SPC]PARA[SPC]FIN
."
170 GETD$:IFD$=""THEN170
173 IFD$="F"THENPOKEV,0:POKESC,6:POK
EBO,14:PRINT"[CLR][COMM7]":END
175 D=VAL(D$)
180 IFD>3ORD<1THEN170
200 PRINT"[CLR][RED]":R=0:W=0:POKESC
,8:POKEBO,2
210 PRINTTAB(12)"[15COMM@]":PRINTTAB
(10)"[RVSON][SHIFT£][RVSOFF][SHIFT£]
";SPC(15);"[SHIFTM]"
211 PRINTTAB(9)"[RVSON][SHIFT£][RVSOFF]
[SHIFT£][SPC]11[5SPC]12[6SPC]1[SHIFTM]
":PRINTTAB(8)"[RVSON][SHIFT£][RVSOFF]
[SHIFT£]";SPC(19);"[SHIFTM]"
212 PRINTTAB(8)"[RVSON][SPC][RVSOFF]
10";SPC(18);"2[SHIFTM]":PRINTTAB(8)"
[RVSON][SPC][RVSOFF]";SPC(22);"[COMMG]
"
213 PRINTTAB(8)"[RVSON][SPC][RVSOFF]
";SPC(22);"[COMMG]":PRINTTAB(8)"[RVSON]
[SPC][RVSOFF]";SPC(22);"[COMMG]"
214 PRINTTAB(8)"[RVSON][SPC][RVSOFF]
";SPC(22);"[COMMG]":PRINTTAB(8)"[RVSON]
[SPC][RVSOFF]";SPC(22);"[COMMG]"
215 PRINTTAB(8)"[RVSON][SPC][RVSOFF]
";SPC(9);"[RVSON][4SPC][RVSOFF]";SPC
(9);"[COMMG]"
216 PRINTTAB(8)"[RVSON][SPC][RVSOFF]
[SPC]9";SPC(7);"[RVSON][4SPC][RVSOFF]
";SPC(7);"3[SPC][COMMG]"
217 PRINTTAB(8)"[RVSON][SPC][RVSOFF]
";SPC(9);"[RVSON][4SPC][RVSOFF]";SPC
(9);"[COMMG]"
218 PRINTTAB(8)"[RVSON][SPC][RVSOFF]
";SPC(9);"[RVSON][4SPC][RVSOFF]";SPC
(9);"[COMMG]"
219 PRINTTAB(8)"[RVSON][SPC][RVSOFF]
";SPC(22);"[COMMG]":PRINTTAB(8)"[RVSON]
[SPC][RVSOFF]";SPC(22);"[COMMG]"
220 PRINTTAB(8)"[RVSON][SPC][RVSOFF]
";SPC(22);"[COMMG]":PRINTTAB(8)"[RVSON]
[SPC][RVSOFF]";SPC(22);"[COMMG]"
221 PRINTTAB(8)"[RVSON][SPC][RVSOFF]
";SPC(22);"[COMMG]":PRINTTAB(8)"[COMM*]
[RVSON][COMM*][RVSOFF]8";SPC(18);"4[SHIFTN]
"
222 PRINTTAB(9)"[COMM*][RVSON][COMM*]
[RVSOFF]";SPC(18);"[SHIFTN]":PRINTTA
B(10)"[COMM*][RVSON][COMM*][RVSOFF]7
";SPC(6);"6";SPC(7);"5[SHIFTN]"
223 PRINTTAB(11)"[COMM*][RVSON][COMM*]
[RVSOFF][14COMM@][SHIFTN]"
240 HR=INT(RND(1)*12+1):0=0
24[illegible]=3THENMN=INT(RND(1)*12):GOTO2
48
243 IFD=2THENMN=6*INT(RND(1)*2):GOTO
248
247 MN=0
248 IFMN=0ORMN=1THENPRINT"[HOM]";HR;
":0";5*MN:GOTO250
249 PRINT"[HOM]";HR;":";5*MN
250 H=1:RM=0
251 IFRM>2THENPL=VAL(MID$(LH$,(H-1)*
4+1,4)):GOTO280
261 IFRM>11THENH=H+1:RM=0
262 IFH>12THEN250
270 PL=VAL(MID$(L$,(H-1)*4+1,4))
280 PB=VAL(MID$(B$,4*RM+1,4))
284 IFRM=0THENII=40:PP=66:GOTO290
285 II=VAL(MID$(IP$,(RM-1)*5+1,3))
286 PP=VAL(MID$(IP$,(RM-1)*5+4,2))
289 IF(RM>11)OR(H>12)THEN261
```

```
230 POKEPB,160:POKEPB+CO,0
300 FORK=1TO5:POKEPB+K*II,PP:POKEPB+
K*II+CO,0:NEXTK
305 POKEH2,30:POKEL2,141:POKEW2,17:F
ORK=1TO10:NEXTK:POKEW2,0
310 IFRM>2THENI=VAL(MID$(IH$,(H-1)*5
+1,3)):P=VAL(MID$(IH$,(H-1)*5+4,2))
GOTO330
320 I=VAL(MID$(IP$,(H-1)*5+1,3)):P=V
AL(MID$(IP$,(H-1)*5+4,2))
330 POKEPL,160:POKEPL+CO,6
340 FORK=1TO3:POKEPL+K*I,P:POKEPL+K*
I+CO,6:NEXTK
350 FORK=1TO600:NEXTK
353 IFQ2=1THEN650
355 GETS$:IFS$=""THEN379
357 IFS$="[F1]"THENRM=RM+12:GOTO379
358 IFS$="[F3]"THENRM=RM+2:GOTO379
359 IFHR=HANDMN=RMTHEN500
360 GOTO600
[illegible]OKEH2,61:POKEL2,126:POKEW2,17:F
ORK=1TO10:NEXTK:POKEW2,0
380 POKEPB,32:FORK=1TO5:POKEPB+K*II,
32:NEXTK
381 POKEPL,32:FORK=1TO3:POKEPL+K*I,3
2:NEXTK
385 IFQ=1THEN240
386 IFQ2=1THEN251
390 RM=RM+1
391 IFRM>11THENH=H+1:RM=0
392 IFH>12THEN250
400 GOTO251
```

```
500 FORK=1TO2:PRINT"[HOM]"SPC(20);"S I"
501 FORX=1TO10:POKEH2,108:POKEL2,223
:POKEW2,17:FORJ=1TO10:NEXTJ
502 POKEW2,0:NEXTX
503 PRINT"[HOM]";SPC(20);"[4SPC]":FO
RX=1TO100:NEXTX:NEXTK
504 R=R+1:W=0:PRINT"[HOM]";SPC(37);R

505 IFR=3THENPRINT"[HOM][SPC][RVSON]
FANTASTICO[RVSOFF][10SPC]":GOTO507
506 PRINT"[HOM]PULSA[SPC]'F7'[SPC]PA
RA[SPC]EMPEZAR.":GOTO509
507 FORX=1TO75:POKEW2,33:POKEH2,X:PO
KEL2,100+2*X:FORJ=1TO10:NEXTJ:NEXTX
508 POKEW2,0:GOTO700
509 GETS$:IFS$=""THEN509
510 PRINT"[HOM][24SPC]":Q=1:GOTO380
600 POKEH2,8:POKEL2,23:POKEW2,33:FOR
X=1TO900:NEXTX:POKEW2,0
601 Q2=1:RM=MN:H=HR:W=W+1
603 GOTO380
650 PRINT"[HOM]";SPC(9);"CORREGIDO":
Q2=0
652 FORX=1TO2000:NEXTX
653 IFW>2THEN10
654 PRINT"[HOM]PULSA[SPC]'F7'[SPC]PA
RA[SPC]EMPEZAR.[2SPC]"
655 GETS$:IFS$=""THEN655
657 PRINT"[HOM][26SPC]"
660 Q=1:GOTO380
700 FORX=1TO1000:NEXTX
701 GOTO150
```

FIN

COMO RESUCITAR LOS FICHEROS ASESINADOS

No es necesario que intervenga la mano de Dios para que se resuciten los ficheros asesinados

En ese momento hubiera tirado el ordenador por la ventana con mucho gusto y encima hubiera disfrutado enormemente al ver cómo se hacía pedazos en el suelo allí abajo. Me daba igual que la metedura de pata había sido mía, el ordenador ni siquiera había intentado impedírmelo.

Acababa de borrar las únicas dos copias que tenía de un programa que me había costado cinco días de trabajo. No me preguntes por qué lo hice; en ese momento yo tenía los cables cruzados. Al cargar y listar el directorio del disco para buscar mi programa, descubrí horrorizado que ya no estaba allí. Empecé a sudar.

Pero entonces me paré a pensar. Si un fichero se borra, no quiere decir que se haya destruido. Los registros físicos todavía están en el disco. Lo único que ha cambiado ha sido la entrada en el directorio del disco. Si se pudiera volver a cambiar la entrada, el fichero podía ser restaurado siempre y cuando no se volviera a salvar un programa nuevo en el mismo disco.

Con mucha calma, saqué el disco de la unidad, lo guardé en un lugar seguro, y formateé un disco nuevo para reemplazarlo. Al consultar el manual de la unidad de discos 1541 descubrí lo que quería saber: un solo byte de datos en el directorio del disco marca la diferencia entre un fichero "vivo" y un fichero "muerto". Si pudiese devolverle el valor correcto al byte, el fichero volvería a aparecer milagrosamente.

C-64
UNIDAD DE DISCOS 1541



El problema era sacar el byte del disco, modificarlo y volverlo a colocar. Los comandos Read-Block y Write-Block (o USER1 y USER2) podían ser utilizados para sacar el bloque y volver a meterlo. Sin embargo, otra cuestión sería realizar una modificación en dicho bloque.

Lo que estaba claro era que me hacía falta un programa editor de discos bastante complicado. Como no encontré ninguno, decidí escribirlo yo mismo.

El programa "Editor de Discos 64" es un editor de pistas y sectores de pantalla entera para el C-64 y la unidad de discos 1541. Cualquier pista y sector —es decir, un bloque— puede ser llamado, modificado y devuelto al disco. Además, cualquier modificación realizada pero no grabada puede ser borrada para poder empezar de nuevo, si es necesario.

Geografía del Disco

Formatear un disco significa que la superficie del disco queda dividido en una serie de pistas (unas anillas magnéticas concéntricas sobre las cuales se graba la información codificada), y sectores (los segmentos de las anillas). La información almacenada en un sector de una sola pista se llama un bloque.

Un bloque, que puede contener hasta 256 bytes de información, es la unidad más pequeña donde los datos pueden ser almacenados en un disco. Una secuencia lógica de bloques, que contiene un solo programa, o una serie de registros, o cualquier otra cosa, constituye un fichero. Los bloques que componen el fichero no se encuentran necesariamente en orden, pero el sistema operativo de disco (DOS) de Commodore los trata como si lo estuvieran.

Cuando se carga un programa del disco, el DOS consulta el directorio del disco, localiza el bloque donde comienza un fichero, determinado y transfiere cada bloque del fichero, en la secuencia correcta, a la memoria del ordenador, como si el programa se hubiese almacenado como una sola unidad consecutiva. Los bloques dentro de un fichero siempre son identificados, en el directorio y en cualquier otro sitio, por sus números de pista y de sector. Después hablaremos de esto con más detalle.

Lo primero que hace el programa "Editor de Discos 64" es solicitar un número de pista. En la unidad 1541, este número puede ser cualquiera entre 1 y 35. A continuación pide un número de sector más alto para una pista determinada. La figura 1 muestra el número de sectores en cada pista.

Se puede contestar con números decimales o hexadecimales dentro del rango permitido. (Los números hexadecimales son identificados por el signo de dólar que llevan delante ($0011)). Los números que se encuentren fuera del límite permitido no serán aceptados.

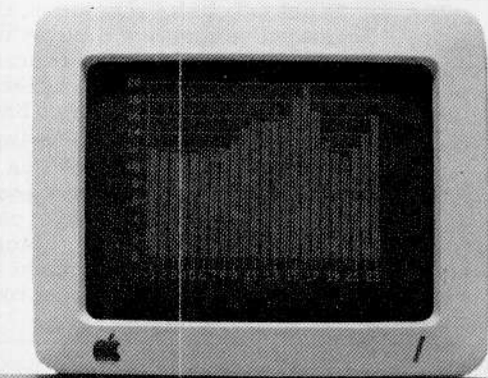
Si se elige, por ejemplo, la pista 10 y el sector 4, aparece en pantalla el mensaje LEYENDO PISTA 10 SECTOR 4 PARTE

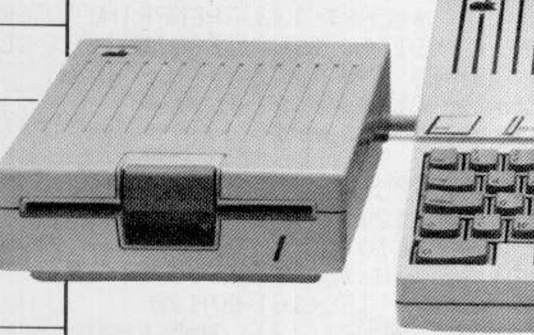
Suscríbase a



y le llevaremos a su casa un APPLE II c

Suscríbase a MicroSistemas antes del 30 de noviembre próximo y participará en el sorteo de un magnífico ordenador personal APPLE II c.
Para los que ya estén suscritos también tendrán opción de conseguir el APPLE II c. Si desea conocer su número en el sorteo llámenos al teléfono 231 23 95.
Habrá también otros premios para nuestros suscriptores que daremos oportunamente a conocer. El sorteo tendrá lugar en nuestro stand del SIMO.







BOLETIN DE SUSCRIPCION A MICROSISTEMAS

Nombre y Apellidos ....
Puesto ....
Compañía ....
Actividad Empresa ....
Dirección .... Teléf.: ....
Ciudad .... D. P. .... Provincia ....

Suscripción anual 2.950 ptas.
Deseo iniciar la suscripción con el próximo número
☐ Adjunto cheque por 2.950 ptas.

☐ Reembolso más gastos del mismo al recibir el primer número de la suscripción.
(Enviar a Gravina, 13-1º Izda. Madrid-4 - Teléf.: 231 23 88/95)

1 ("PARTE 1" se refiere al hecho de que sólo la mitad de un bloque —128 bytes— aparece en pantalla a la vez). El contenido del bloque identificado por los números de pista y sector sale en pantalla debajo del mensaje tanto en hexadecimal como en ASCII.

Cuando termina de llenarse la pantalla, aparece un cursor no intermitente en la esquina superior izquierda del volcado hexadecimal, que tapa por completo el primer par de dígitos hexadecimales. Puedes mover el cursor utilizando las teclas normales para el desplazamiento del cursor y colocarlo encima de los dígitos hexadecimales que quieres modificar. Para realizar el cambio tecleas los dígitos nuevos.

Recuerda que es obligatorio modificar dos dígitos a la vez. Es decir, una vez tecleado un solo dígito hexadecimal, hay que teclear el segundo antes de poder mover el cursor. Esto puede parecer un poco pesado pero te acostumbrarás rápido.

Los caracteres ASCII no se actualizan al realizar una modificación. Es necesario volver a cargar el bloque una vez terminadas las modificaciones para poder ver los cambios realizados en el formato ASCII.

Cuando terminas de hacer los cambios en la parte 1 del bloque, pulsa la tecla Return, el bloque modificado se grabará en el disco y la parte 2 saldrá en pantalla. Al completarse la segunda parte, vuelve a pulsar la tecla Return; el bloque será grabado y se pasará automáticamente al siguiente bloque en orden secuencial (en vez de pasar al siguiente bloque del fichero).

No se realiza ningún cambio en el disco hasta que no se pulsa la tecla Return. Si, antes de pulsar la tecla Return, decides no grabar las modificaciones realizadas, pulsa la tecla flecha arriba. (Esta es la tecla que lleva la flecha arriba y no la tecla de cursor arriba). La pantalla será borrada, tus modificaciones desaparecerán, y el programa te pedirá otra pista y otro sector.

El programa "Editor de Discos 64" también puede ser utilizado para leer y examinar las pistas y los sectores del disco, sin tener que realizar ningún cambio. Siempre que se pulsa la tecla Return el programa avanza al próximo sector (y a la siguiente pista si la anterior se está finalizando), o a la segunda parte del sector actual.

Si pulsas la tecla flecha arriba, el programa vuelve a solicitar la pista y sector. O si no, pulsando la tecla F1 se finaliza el programa, se cierran todos los ficheros, y se vuelve al modo inmediato de Basic.

Un aviso: al probar este programa por primera vez, no lo hagas con un disco bueno. Es mejor formatear un disco nuevo y crear unos ficheros falsos para la prueba. Un error al teclear el programa podría resultar en la destrucción de un bloque entero, y si dicho bloque contiene la cabecera del directorio del disco, se podrían perder todos los ficheros del disco.

Para que esto no ocurra, lee el programa con mucho cuidado y utilízalo para realizar unas modificaciones de poca importancia en un bloque no usado. (Los bloques no usados normalmente contienen los caracteres hexadecimales 01). Graba estas modificaciones en disco, pasando al bloque siguiente, pulsa la tecla flecha arriba, y vuelve a mirar el primer bloque de nuevo.

Si las modificaciones —y todo lo demás, excepto los caracteres ASCII actualizados— siguen igual, significa que el programa funciona bien. Si no, vuelve a repasar el programa. No utilices el "Editor de Discos 64" con un disco bueno hasta que no tengas la seguridad de que lo hayas tecleado correctamente.

Ejemplo

A continuación vamos a ver un ejemplo de cómo funciona el programa. Carga el programa, teclea RUN, e introduce el disco que quieres editar en la unidad de discos 0.

En general, el primer bloque que se examina es el que contiene el directorio: Pista 18, Sector 1. Esto te indica todos los ficheros guardados en el disco, donde se encuentran y el número de bloques que ocupan.

Consulta el capítulo 5 del "Manual del Usuario" de la Unidad de Discos 1541

```
10 DIMS%(35),B$(1):PRINTCHR$(147)CHR
$(151);:POKE53280,1:POKE53281,1:GOSU
B680
20 OPEN15,8,15:OPEN5,8,5,"#"
30 PRINTCHR$(147);:N=11:CH=17:GOSUB7
00:N=3:CH=29:GOSUB700:PRINT"TRACK";:
INPUTT$
35 N=3:CH=29:GOSUB700:PRINT"SECTOR";
:INPUTS$:X=0:TF=0:SF=0
40 IFLEFT$(T$,1)="$"THENHX$=T$:GOSUB
480:T=DC:TF=1
50 IFLEFT$(S$,1)="$"THENHX$=S$:GOSUB
480:S=DC:SF=1
60 IFSF=0THENS=VAL(S$)
70 IFTF=0THENT=VAL(T$)
80 IFT<1ORT>35THEN30
90 IFS<0ORS>S%(T)THEN30
100 GOSUB350
110 PRINTCHR$(19);:N=5:CH=17:GOSUB70
0:N=6:CH=29:GOSUB700:C=0:L=0
120 PRINTCHR$(18)CHR$(32)CHR$(32)CHR
$(146)CHR$(157)CHR$(157);
125 DC=ASC(MID$(B$(X),L*8+C+2,1)+CHR
$(0)):GOSUB450
130 GETK$:IFK$=""THEN130
140 IFK$=CHR$(29)THEN230
150 IFK$=CHR$(17)THEN260
160 IFK$=CHR$(145)THEN290
170 IFK$=CHR$(157)THEN320
180 IFK$=CHR$(13)THEN590
190 IF(K$=>"0"ANDK$=<"9")OR(K$=>"A"A
NDK$=<"F")THEN530
200 IFK$=CHR$(94)THEN30
210 IFK$=CHR$(133)THENPRINT"[CLR]";:
PRINT"EDITING[SPC]TERMINATED":CLOSE1
5:CLOSE5:END
220 GOTO130
230 PRINTHX$CHR$(29);:C=C+1:IFC<8THE
N120
240 N=16:CH=29:GOSUB700:L=L+1:C=0:IF
L<16THEN120
250 GOTO110
260 PRINTHX$CHR$(157)CHR$(157)CHR$(1
7);:L=L+1:IFL<16THEN120
270 PRINTCHR$(19);:N=5:CH=17:GOSUB70
0:N=6:CH=29:GOSUB700
275 IFC>0THENFORJ=1TOC*3:PRINTCHR$(2
9);:NEXT
280 L=0:GOTO120
290 PRINTHX$CHR$(157)CHR$(157)CHR$(1
45);:L=L-1:IFL=>0THEN120
300 PRINTCHR$(19);:N=20:CH=17:GOSUB7
00:N=6:CH=29:GOSUB700
305 IFC>0THENFORJ=1TOC*3:PRINTCHR$(2
9);:NEXT
310 L=15:GOTO120
320 PRINTHX$;:N=5:CH=157:GOSUB700:C=
C-1:IFC=>0THEN120
330 C=7:L=L-1:N=16:CH=157:GOSUB700:I
FL=>0THEN120
340 L=15:PRINTCHR$(19);:N=20:CH=17:G
OSUB700:N=27:CH=29:GOSUB700:GOTO120
```

para saber el formato específico del almacenamiento de datos. Sin embargo, la información contenida en el manual es, a veces, muy confusa y no conviene hacerle demasiado caso, usa el manual simplemente como guía para tu propia exploración del disco.

Cuando el programa te pide un número de pista, teclea 18 (o $12, en hexadecimal) y pulsa Return. Cuando te pide un número de sector, teclea 1 (o $01 en hexadecimal) y pulsa Return. El mensaje LEYENDO PISTA 18 SECTOR 1 PARTE 1 aparecerá en pantalla. La primera mitad del bloque sale en pantalla debajo del mensaje.

Se pueden leer los nombres de todos los ficheros almacenados en el disco en el volcado ASCII. Los tres bytes hexadecimales que aparecen delante del primer byte del nombre del programa son muy importantes. El primer byte de los tres indica el estado del fichero, de acuerdo con la siguiente tabla:

$00 - Borrado	$83 - Usuario
$81 - Secuencial	$84 - Relativo
$82 - Programa	

Cuando empecé a recuperar mi fichero "asesinado", el primer problema era transformar el byte que indica el estado del fichero del primero al tercero —es decir, cambiar el "00" del fichero borrado al "82" del fichero activo. Con el programa "Editor de Discos" esta operación es muy sencilla.

Cuando el cursor aparece en la esquina superior izquierda, utiliza las teclas del cursor para situarlo encima del tercer byte (que debe ser 00) antes del nombre del fichero "asesinado", y teclea el número 82. Como por arte de magia, el fichero será resucitado y devuelto al directorio, y puede ser cargado como cualquier otro fichero, siempre y cuando ningún fichero nuevo haya sido grabado encima.

Los siguientes dos bytes antes del nombre contienen los números de la pista y sector del bloque donde empieza el fichero. Si quieres mirar un fichero determinado, apunta estos números, pulsa la tecla flecha arriba, e indica al programa "Editor de Discos" que te enseñe este bloque. (Recuerda que estos números de pista y sector son representados en hexadecimal, y deben introducirse con el signo $ delante, o traducidos a decimal).

Los dos primeros bytes de un fichero de programa contienen los números de pista y sector del siguiente bloque en el fichero. El tercero y el cuarto byte contienen —en orden ascendente, es decir, al reves del orden normal de lectura— la dirección de carga para el programa, es decir, la dirección de memoria donde el fichero será almacenado en la memoria, si se ha cargado con el comando LOAD 8,1. (Para programas en Basic, esta dirección normalmente es $0801, la dirección por falta para todas las cargas de programas).

Si perdieras la dirección SYS para un programa en lenguaje máquina, puedes emplear el programa "Editor de Discos" para localizar la dirección de carga, dado que la dirección del SYS y la de carga normalmente, pero no siempre, son las mismas. Si quieres, puedes modificar los bytes para cambiar la dirección de carga para el programa. Sin embargo, recuerda que la mayoría de los programas en lenguaje máquina no se ejecutarán correctamente si cambian de posición.

Los bytes restantes del fichero contienen el programa, exactamente de la misma forma en que se almacenó en memoria. Si el programa está en Basic, esto significa que todas las palabras clave están codificadas —es decir, ocupan un solo byte— y cada línea de la programación termina en cero y va precedida por cuatro bytes que indican la posición de memoria de la línea siguiente y el número de la línea actual.

Cuando terminas la exploración o modificación de tu disco, y el último bloque ha sido grabado en disco, pulsa F1 y volverás al modo inmediato.

Figura 1. Sectores por pista.

Número de pista	Número de Sector Más Alto
1-17	20
18-24	18
25-30	17
31-35	16

```
350 PRINTCHR$(147)CHR$(17)"READING
[SPC]TRACK"T"[SPC]SECTOR"S"[SPC]PART
"X+1:N=3
355 CH=17:GOSUB700
360 B$(X)="":PRINT#15,"U1:"5;0;T;S
365 PRINT#15,"B-P:"5,0
370 IFX=1THENFORI=1TO128:GET#5,A$:NE
XT:GOTO380
375 B$(1)=""
380 FORJ=0TO15:DC=J*8+128*X:GOSUB450
:PRINT"00"HX$"[2SPC]";:FORK=0TO7
390 GET#5,A$:A$=CHR$(ASC(A$+CHR$(0))
):B$(X)=B$(X)+A$
400 DC=ASC(A$):GOSUB450:PRINTHX$"
[SPC]";:NEXT:PRINT"[SPC]";:FORK=0TO7
410 DC=ASC(MID$(B$(X),J*8+K+1,1)+CHR
$(0))
420 IF(DC>32ANDDC<128)ORDC>159THENPR
INTCHR$(DC);:GOTO440
430 PRINT".";
440 NEXT:PRINT:NEXT:B$(X)="[SPC]"+B$
(X)+"[SPC]":RETURN
450 D1=INT(DC/16):D2=DC-D1*16
460 HX$=CHR$(D1+48-7*(D1>9))+CHR$(D2
+48-7*(D2>9)):RETURN
470 HX$=RIGHT$(HX$,LEN(HX$)-1):IFLEN
(HX$)=0THENHX$="00"
480 IFLEN(HX$)=1THENHX$="0"+HX$
490 IFLEN(HX$)>2THENHX$=RIGHT$(HX$,2
)
500 FORI=1TO2:A$=MID$(HX$,I,1)
505 IFA$<"0"OR(A$>"9"ANDA$<"A")ORA$>
"F"THENDC=0:RETURN
510 K$=LEFT$(HX$,1):J$=RIGHT$(HX$,1)

520 D1=ASC(K$)-48:D2=ASC(J$)-48:DC=(
D1+7*(D1>9))*16+(D2+7*(D2>9)):RETURN
530 PRINTK$"[CRSRL]";
540 GETJ$:IFJ$=""THEN540
550 IFJ$<"0"OR(J$>"9"ANDJ$<"A")ORJ$>
"F"THEN540
560 HX$=K$+J$:GOSUB520
570 B$(X)=LEFT$(B$(X),L*8+C+1)+CHR$(
DC)+RIGHT$(B$(X),LEN(B$(X))-(L*8+C+2
))
580 GOTO230
590 B$(X)=MID$(B$(X),2,LEN(B$(X))-2)
600 PRINT#15,"B-P:"5,0
610 PRINT#5,B$(0);B$(1);
620 PRINT#15,"U2:"5;0;T;S
630 IFX=0THENX=1:GOTO100
640 X=0:GOSUB650:GOTO100
650 S=S+1:IFS<=S%(T)THENRETURN
660 S=0:T=T+1:IFT<36THENRETURN
670 T=1:RETURN
680 FORI=1TO17:S%(I)=20:NEXT:FORI=18
TO24:S%(I)=18:NEXT
690 FORI=25TO30:S%(I)=17:NEXT:FORI=3
1TO35:S%(I)=16:NEXT:RETURN
700 FORII=1TON:PRINTCHR$(CH);:NEXT:R
ETURN
```

FIN

DESCRIPCIÓN ALFABÉTICA DE LOS MNEMÓNICOS DEL 6502/6510 (VI)

INX — INX

Incrementa el registro X en una unidad

Operación: X + 1 → X

(Ref.: 7.4)

N Z C I D V
√ √ – – – –

Modo de Direc.	Formato en ensamblador	Código Operan.	Núm. Bytes	Núm. Ciclos
Implícito	INX	E8	1	2

INY — INY

Incrementa el registro y en una unidad

Operación: Y + 1 → Y

(Ref.: 7.5)

N Z C I D V
√ √ – – – –

Modo de Direc.	Formato en ensamblador	Código Operan.	Núm. Bytes	Núm. Ciclos
Implícito	INY	C8	1	2

JMP — JMP

Salta a una nueva dirección

Operación: (PC + 1) → PCL
(PC + 2) → PCH

(Ref.: 4.0.2 y 9.8.1)

N Z C I D V
– – – – – –

Modo de Direc.	Formato en ensamblador	Código Operan.	Núm. Bytes	Núm. Ciclos
Absoluto Indirecto	JMP Oper. JMP (Oper.)	4C 6C	3 3	3 5

JSR — JSR

Salta a nueva dirección guardando dirección de retorno

Operación: PC + 2 ↓ , (PC + 1) → PCL
(PC + 2) → PCH

(Ref.: 8.1)

N Z C I D V
– – – – – –

Modo de Direc.	Formato en ensamblador	Código Operan.	Núm. Bytes	Núm. Ciclos
Absoluto	JSR Oper.	20	3	6



HES-MON 64 versus EXTRAMON

Por Jordi SASTRE

En este número comenzamos a publicar el manual de utilización del cartucho HES-MON 64 para el COMMODORE 64. Ya existe, desde hace tiempo, un utilitario muy similar para las series CBM 4000 y CBM 8000 con el nombre de EXTRAMON. Mi propósito es comentar las diferencias entre ambos programas. Las diferencias son muy pocas y no dan lugar a un artículo completo, por lo que este escrito, puede considerarse como una nota suplementaria al manual.

El cometido, manejo, limitaciones, posibilidades y ventajas son casi idénticas en ambos utilitarios. Aquí voy a enumerar y detallar cuáles son estas diferencias, para que el conocedor de uno también pueda trabajar con el otro y viceversa.

El EXTRAMON tiene los siguientes comandos que el HES MON no posee:

E.—(Enable Stop Key). Permite interrumpir la ejecución de programas en código máquina pulsando STOP y "=" en el CBM-4032, o STOP y "←" en el CBM-8032. El programa tiene que haberse puesto en marcha mediante el comando G (Go Run).

U.—(Undo Stop Key). Impide la interrupción de programas en código máquina. Comando contrario a E.

Por su parte, el HES MON incorpora los siguientes comandos que el EXTRAMON no tiene:

E.—(External Relinker). Facilita la adaptación al C-64 de programas escritos en otras máquinas.

O.—(Output Divert). Desvía la salida de pantalla a otro periférico, como el disco o la impresora.

P.—(Print Screen). Efectúa en la impresora una copia de contenido actual de la pantalla.

U.—(Test Color RAM). Efectúa un test de la RAM de color.

V.—(Verify RAM). Efectúa un test de un bloque de memoria RAM.

#.—Conversión decimal a hexadecimal.

$.—Conversión hexadecimal a decimal.

+.—Suma hexadecimal.

-.—Resta hexadecimal.

Además de los comandos, EXTRAMON y HES MON tienen también diferencias en cuanto a presentación. Mientras que el HES MON 64 está contenido en cartucho, el EXTRAMON se presenta en una EPROM de 4K que debe ubicarse en el zócalo UD12 de la CPU (dirección $9000).

El EXTRAMON se pone en marcha ejecutando: SYS (36864).

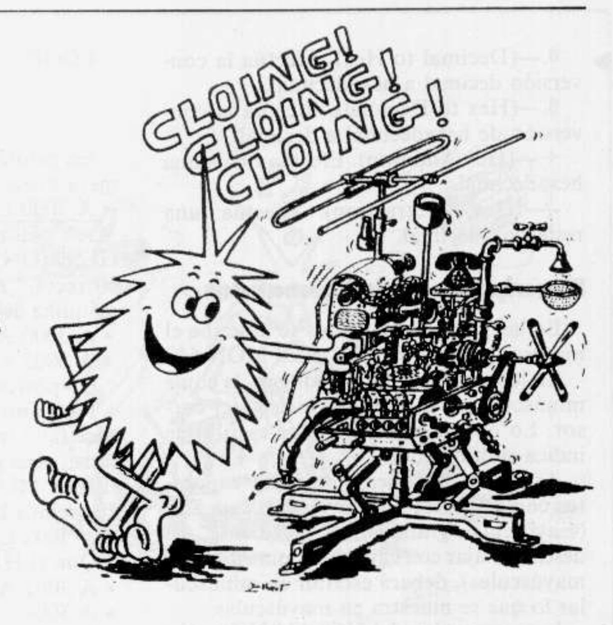
Aunque esté puesta la EPROM, si no se ejecuta el SYS el EXTRAMON no se activa, por lo que mientras tanto no interfiere la CPU.

Cuando se ha puesto en marcha una vez, el comando X permite volver al Basic. Si se desea activar de nuevo el EXTRAMON basta con ejecutar SYS 4 (llamada al MLM residente, que está interferido por el EXTRAMON). En el C-64, para volver a entrar en el HES MON debe pulsarse la tecla RESTORE.

Las demás diferencias que puedan haber son las propias del ordenador (C-64, CBM-4032, CBM-8032), que cada usuario ya conoce y no vale la pena enumerar.



MANUAL DEL MONITOR DE CODIGO MAQUINA PARA EL C-64 HES-MON (I)



HES-MON 64 es un Monitor de Lenguaje Máquina (Machine-Language Monitor). El MLM sirve básicamente para alterar el contenido de los bytes de memoria, leerlos y grabarlos sobre periféricos (cassette o disco) y ordenar la ejecución de un programa en código máquina existente en RAM. HES-MON 64, además de estas funciones, incorpora otras como el desensamblado del código máquina en assembler, programación en lenguaje ensamblador directamente sobre la RAM (sin manejo de etiquetas), rellenado de porciones de RAM con un carácter, relocatación de programas o tablas en código máquina de un lugar a otro de la RAM actualizando, o no, las direcciones de las bifurcaciones absolutas (JMP y JSR), asignación de puntos de ruptura en los lugares que interese de rutinas en código máquina, posibilidad de ejecutar programas en código máquina "paso a paso", visualizando en la pantalla el comando en assembler que es está ejecutando, etc. Como se ve, es una herramienta vital para el que hace sus programas en código máquina, lenguaje cuya depuración sería tarea ardua y difícil si no fuera por utilitarios como éste. Es necesario aclarar para aquellos de nuestros lectores que no conozcan a fondo la programación en código máquina que aunque este cartucho permite entrar programas en el lenguaje propio del microprocesador no sustituye a un assembler para generar grandes y complejos programas, sino que más bien lo complementa al facilitar las tareas de "puesta en marcha" de programas escritos con otros compiladores. Otra de sus utilidades es la de permitir el desarrollo de pequeñas rutinas en código máquina que sustituirán a segmentos de BASIC en lugares donde el tiempo de ejecución sea crítico.

Activación

HES-MON 64 está contenido en un cartucho que deberá conectarse en el port de expansión del COMMODORE 64.

Se activa automáticamente al dar tensión al ordenador.

Desde su activación visualiza el valor de los registros del microprocesador, contador de programa (PC), acumulador, índices X e Y, y puntero del stack.

Por ejemplo:

```
         HESMON 2.0
      BY T.M. PETERSON
       (C) 1983 H.E.S.

C*
PC   IRQ  SR  AC  XR  YR  SP
; 0000 EA31  27  00  00  05  FA
```

Los valores de cada uno de los registros pueden variar dependiendo del momento en que se active el HES MON 64.

Una vez activado, quedan habilitados todos los comandos que posee, que se describen en las páginas siguientes.

Comandos del HES-MON 64

A continuación se describen los comandos del HES MON 64:

A.—(Simple Assembler). Introducción de programas en lenguaje ensamblador directamente sobre RAM.

B.—(Break Set). Asigna "breaks" (puntos de ruptura) en lugares concretos del programa para que la ejecución se interrumpa al pasar cierto número de veces por ellos.

D.—(Disassembler). Visualiza el contenido en código máquina de la RAM o de la ROM.

E.—(External Relinker). Permite la conversión automática de direcciones para adaptar programas de otras máquinas al CBM-64.

F.—(Fill Memory). Rellena porciones de memoria con cierto contenido.

G.—(Go Run). Inicia la ejecución de programas en código máquina a partir de una dirección concreta.

H.—(Hunt Memory). Busca valores concretos a lo largo de la zona de memoria indicada.

I.—(Interpret Memory). Visualiza el contenido de la memoria en hexadecimal y en caracteres ASCII.

L.—(Load). Carga en RAM programas contenidos en disco o cassette.

M.—(Memory display). Visualiza el contenido de la memoria en hexadecimal.

N.—(New Locator). Traslada código máquina o tablas de un lugar a otro de la memoria.

O.—(Output Divert). Desvía la salida de pantalla a otro periférico; p.e. impresora o disco.

P.—(Print Screen). Efectúa una copia del contenido de la pantalla en la impresora.

Q.—(Quick Trace). Ejecuta un programa a partir de la dirección indicada. Permite interrumpir la ejecución.

R.—(Register Display). Visualiza el contenido de los registros del mismo modo que cuando se activa el HES-MON 64.

S.—(Save). Graba el contenido de la memoria sobre disco o cassette.

T.—(Transfer Memory). Transfiere contenido de memoria de un lugar a otro de la RAM.

U.—(Test Color RAM). Ejecuta un test de la RAM de color.

V.—(Verify RAM). Ejecuta un test de un bloque de memoria RAM.

W.—(Walk Mode). Ejecuta "paso a paso" un programa en código máquina.

X.—(Exit to Basic). Sale del HES MON 64 y retorna al Basic. El HES-MON 64 puede activarse de nuevo pulsando la tecla RESTORE. ▶



#.—(Decimal to Hex). Efectúa la conversión decimal a hexadecimal.

$.—(Hex to Decimal). Efectúa la conversión de hexadecimal a decimal.

+.—(Hex Addition). Efectúa una suma hexadecimal.

-.—(Hex Subtraction). Efectúa una resta hexadecimal.

Descripción de los comandos

En las páginas siguientes se describe el manejo de los comandos HES MON 64.

En los ejemplos que se pongan, la equis minúscula (x) indica la posición del cursor. Lo que el operador debe escribir se indica subrayándolo.

Asimismo, los ejemplos están expuestos como si la CPU estuviera en caja alta (mayúsculas/gráficos); lógicamente, si desea trabajar con caja baja (minúsculas/-mayúsculas), deberá escribir en minúsculas lo que se muestra en mayúsculas.

Los comandos del HES MON 64 sólo funcionan cuando el HES MON 64 está activado. Su formato consiste en una letra o un símbolo, que identifican al comando, seguida de los parámetros. Entre el comando y los parámetros tiene que haber un espacio. Los parámetros deben introducirse en formato hexadecimal de cuatro dígitos sin el signo dólar ($) que normalmente identifica las notaciones hexadecimales (0400, B000, 08FA, FCC0, 000C...) a excepción de los comandos FILL MEMORY y HUNT MEMORY que utilizan también parámetros de un byte (dos dígitos hexadecimales: 00, 0B, 12, FF...) como se indica en su descripción correspondiente. Los comandos LOAD y SAVE utilizan nombres de programa; estos deben ser de un máximo de 16 caracteres y deben estar encerrados entre comillas ("NOMBRE PRG", "O: NOMBRE", "1: PEPE"...). El comando SIMPLE ASSEMBLER emplea lenguaje ensamblador; en este caso se aplicarán las reglas del assembler.

Si el HES-MON 64 no entiende un comando por estar mal expresado, bien lo ignora, o bien visualiza un interrogante (?) en la posición donde aproximadamente se encuentra el error. El operador puede escribirlo de nuevo en otra línea, o bien corregir la anterior. Si prueba a ejecutar un comando ilegal, como salvar el propio cartucho en un disco o cinta, también le dará como resultado el interrogante.

A=Ensamblado sencillo

Introducción de programas en lenguaje ensamblador directamente sobre memoria. Por ejemplo, queremos introducir en la posición $3000 (12288 en decimal) el siguiente programa:

```
INICIO    LDX#$00
          LDA#$A0
LOOP      STA $8000, X
          INX
          BNE LOOP
          BRK
```

En primer lugar introduciremos la primera línea:

• A 3000 LDX #$00

• Al pulsar RETURN, puesto que el HES-MON 64 acepta la sintaxis como correcta, traduce la línea a código máquina dejándola de la siguiente manera:

• A 3000 A2 00 LDX #$00

• A 3002 x

(x=posición del cursor)

Para introducir la siguiente línea no hace falta indicar el byte donde debe ubicarse, pues ya lo calcula automáticamente el HES MON 64. Por tanto, ahora introduciremos la línea siguiente:

• A 3002 LDA #$A0

Que el HES-MON 64 convierte en:

• A 3002 A9 A0 LDA #$A0

• A 3004 x

Y así sucesivamente hasta introducir el programa completo, que una vez hecho tendrá el siguiente aspecto:

```
• A 3000 A2 00       LDX#$00
• A 3002 A9 A0       LDA #$A0
• A 3004 8D 00 80    STA $8000,X
• A 3007 E8          INX
• A 3008 DO FA       BNE $3004
• A 300A 00          BRK
• A 300B
• X
```

Para salir del modo SIMPLE ASSEMBLER se ha pulsado RETURN cuando el cursor estaba en la posición $300B. Nótese que en lugar de "BNE LOOP" hemos tenido que poner "BNE $3004", pues el SIMPLE ASSEMBLER no permite las etiquetas.

B=Colocar puntos de ruptura

Comando para situar un punto de ruptura (break-point) en un byte determinado, para que la ejecución del programa se detenga en este punto cuando haya pasado un determinado número de veces por él. Un punto de ruptura, es una marca indicadora de que la ejecución del programa en código máquina debe interrumpirse para pasar el control al MLM, o en este caso al HES MON 64.

Por ejemplo:

• B 1000 00FF

El programa se interrumpirá cuando haya pasado 255 veces ($FF) por la posición de memoria $1000

• B1000

El programa se interrumpirá cuando pase por la dirección $1000 (una sola vez).

C=Comparación de bloques de memoria

Este comando compara dos bloques de memoria y notifica todas las diferencias que encuentra.

Por ejemplo:

• C 1111 2222 3333

Donde 1111 es la primera dirección de la primera sección, 2222 es la última dirección de la primera sección y 3333 es la dirección donde empieza la sección que se comparará con la primera. Este comando se puede detener (en el caso de que se impriman muchas direcciones) con la tecla STOP.

D=Desensamblar

Visualiza una sucesión de direcciones junto con el contenido de sus bytes en hexadecimal, y su equivalente en código máquina.

Por ejemplo:

• D 3000 300B

Provocaría (es un ejemplo) el siguiente desensamblado:

```
., 3000 A2 00       LDX #$00
., 3002 A9 A0       LDA #$A0
., 3004 8D 00 80    STA $8000,X
., 3007 E8          INX
., 3008 DO FA       BNE $3004
., 300A 00          BRK
.x 300B
```

Pueden modificarse directamente los bytes en hexadecimal: el HES MON 64 desensamblará la línea de nuevo. Por ejemplo, para cambiar el valor $8000 de la dirección $3004 por el valor $8150:

La manera es modificando el texto en assembler, es decir...

• A 3004 9D 00 80 STA $8150, X

Y al pulsar RETURN el HES MON 64 actualizará los bytes en hexadecimal:

• A 3004 9D 50 81 STA $8150, X

Si al desensamblar no se especifica dirección final (segundo parámetro) el HES-MON 64 desensamblará hasta el final de la memoria ($FFFF) o hasta que se pulse la tecla STOP

E=Enlace externo

Este comando facilita la adaptación de programas en CODIGO MAQUINA de otras máquinas para el COMMODORE 64. Exactamente este comando cambia las llamadas al sistema operativo del programa fuente por las llamadas propias del sistema operativo del COMMODORE 64. La tabla generada contiene parejas de direcciones que ejecuten en ambos sistemas operativos tareas similares.

Por ejemplo:

• E 1200 13FF 1000 B000 FF00

Los 2 primeros parámetros son el inicio y el fin del programa, el tercero es el inicio de la tabla de direcciones y los dos últimos son el rango de las direcciones en las que queremos hacer cambios.

F=Llenar la memoria

Rellenado de una zona de memoria con un carácter o valor determinado. Por ejemplo:

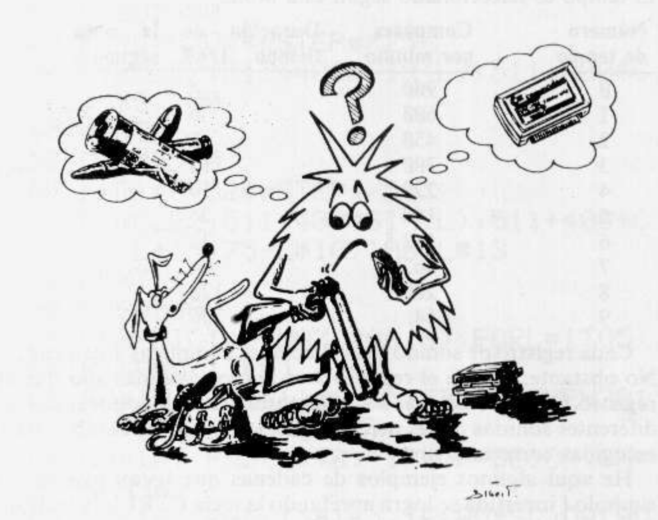
• F 1000 1100 FF

Asigna el valor 255 ($FF) a todos los bytes comprendidos entre $1000 y 1100 (4096 y 4352 en decimal), ambos inclusive■



SEGUNDA Y ULTIMA PARTE

MANUAL DEL SUPER EXPANDER



VIC-20



RPEN(x)

Lee la coordenada de pantalla del lápiz óptico. Si x=0 lee la coordenada X, y si x=1, lee la coordenada Y.

EJEMPLO: X=RPEN(Ø)

RJOY(x)

Lee el valor del joystick. La variable x debe estar entre Ø y 255, y su valor es irrelevante.

EJEMPLO:
```
1Ø GRAPHIC2:COLOR11,6,6,6:X=17Ø:Y=17Ø
2Ø J=RJOY(Ø)
3Ø X=X+((JAND4)=4)-((JAND8)=8)
4Ø Y=Y+((JAND1)=1)-((JAND2)=2)
5Ø POINT3,X*3,Y*3:IFJ=128THEN:SCNCLR
6Ø GOTO2Ø
```

RSND(x)

Lee el valor de cualquiera de los registros de sonido, dependiendo del valor de x según esta tabla:

x	Registro de sonido
1	sonido 1
2	sonido 2
3	sonido 3
4	sonido 4
5	volumen

3. ESCRIBIENDO MUSICA CON EL SUPER EXPANDER

El modo de escribir música de SUPER EXPANDER le permite a usted el poder tocar notas sólo tocando las teclas en el teclado o, de una forma más fácil, usando la sentencia PRINT en sus programas BASIC. Es gobernado por interrupción, lo que permite al programa tocar música mientras está realizando otras tareas. Está pensando principalmente para tocar música, en lugar de efectos de sonido, lo que le permite escribir melodías mucho más fácilmente.

El modo musical es introducido pulsando la tecla CTRL y la flecha a la izquierda (—) al mismo tiempo. Cuando usted lo hace en un programa le aparece una "f inversa". Este es un signo especial que le dice a usted dónde tiene la música en su programa. A partir de este momento, cualquier tecla que toque será interpretada de forma musical, hasta que usted pulse la tecla RETURN. He aquí los caracteres del modo musical:

Carácter	Efecto
P	conecta el eco de pantalla
Q	desconecta el eco de pantalla
V	selección de volumen
S	selección de registro de sonido
O	selección de octava
T	selección de tempo
R	espera el primer compás
C, D, E, F, G, A, B	toca la nota
#	toca la siguiente nota sostenida
$	toca la siguiente nota bemol
RETURN	finaliza el modo musical.

Cuando imprima cadenas de estos caracteres, el punto y coma (;) puede ser usado para impedir que el carácter RETURN acabe con el modo musical.

El eco de pantalla está generalmente desconectado. Cualquiera de los caracteres reconocidos por la caja de música no es mostrado en la pantalla. No obstante, cuando el eco de pantalla está conectado, sí que son mostrados.

Alguna de las letras aceptadas por el modo musical deben ir seguidas por un dígito. He aquí una tabla de esas letras y de sus números asociados:

Letra	Números	Valor de los números
V	Ø a 9	Ø es volumen bajo, 9 es volumen alto
S	1 a 4	número de altavoz
O	1 a 3	1 es bajo, 3 es alto
T	Ø a 9	tempo



El tempo es seleccionado según esta tabla:

Número de tempo	Compases por minuto	Duración de tiempo 1/6Ø	la nota segundos
Ø	9ØØ	4	
1	6ØØ	6	
2	45Ø	8	
3	3ØØ	12	
4	225	16	
5	15Ø	24	
6	112.5	32	
7	56.25	64	
8	28.	128	
9	14.	255	

Cada registro de sonido tiene 3 octavas completas a su alcance. No obstante, ya que el registro 2 es una octava más alta que el registro 1, las tres octavas son diferentes. Cuando toque notas en diferentes sonidos de registro, asegúrese de que las octabas sean escogidas correctamente.

He aquí algunos ejemplos de cadenas que tocan música (el símbolo f invertida se logra apretando la tecla CTRL y la tecla de flecha a la izquierda al mismo tiempo).

4. TECLAS DE FUNCIONES PROGRAMABLES

Hay 8 funciones a las cuales se puede acceder con las teclas de función y la tecla SHIFT. Estas teclas están preasignadas según la siguiente tabla:

Tecla	Graphic
f1	GRAPHIC
f2	COLOR
f3	DRAW
f4	SOUND
f5	CIRCLE
f6	RUN + RETURN
f7	POINT
f8	LIST + RETURN

Toda vez que usted pulse estas teclas, los caracteres para los cuales están programados entrarán en su programa del mismo modo que si usted las hubiese tecleado. Cada tecla puede ser programada hasta con 128 caracteres, usando el comando KEY.

KEY
o
KEY n, "string"

La palabra KEY hace que el VIC le muestre una lista de todos los valores actuales de las teclas de función. Esto está formateado en la pantalla por lo que el contenido de las cadenas puede ser cambiado moviendo el cursor hasta la cadena deseada, haciendo los cambios oportunos y apretando RETURN, como si de un programa normal se tratara. Usted puede cambiar cualquier tecla en el KEY MENU cambiando la información dentro de las marcas de señal y tecleando RETURN o tecleando el comando KEY usando el formato explicado en el ejemplo de abajo.

Cuando sea seguido por un número, este comando cambia el valor de la cadena por el nuevo valor. Este puede ser hasta de 128 caracteres, incluyendo los controles de cursor y de color, el RETURN (CHR$(13)), o cualquier otro carácter. Las teclas pueden ser cambiadas también desde el control de programa.

EJEMPLO: KEY 1, "NO ES MARAVILLOSO?"

5. 3K EXPANSION DE MEMORIA

Cuando se use el cartucho de 3K para expandir la memoria del VIC-20, el área del programa BASIC es cambiada, y pasa a empezar en la localización 1Ø24 ($Ø4ØØ).

Advierta que con el cartucho de 3K conectado, sólo la memoria que está en el cartucho (4Ø96-8191 ó $1ØØØ-1FFF) puede ser usada para la pantalla del VIC y para las áreas de gráficos de alta resolución.

TRES EJEMPLOS DE PROGRAMAS SUPEREXPANDER

Programa 1

```
5 REM TRIG PLOT DEMO
10 GRAPHIC2
11 SCNCLR
15 SC=INT(RND(1)*16):CH=INT(RND(1)*8
):IFSC=CHTHEN15
20 COLORSC,SC,CH,1
30 AM=RND(1)*400+100
40 FR=RND(1)*100+50
50 AD=RND(1)*π*2
60 TA=INT(RND(1)+.3)
70 IFTA=1THENAM=RND(1)*100+50:TR=RND
(1)*200+50:CT=INT(RND(1)+.5)
80 X=0:GOSUB200
90 DRAW1,X,Y:Y0=Y
100 FORX=10TO1024STEP20
110 GOSUB200
120 IFABS(Y-Y0)>600THEN:DRAW1,X,Y
130 DRAW1TOX,Y:Y0=Y
140 NEXT
150 FORX=1TO2000:NEXT
160 RUN
200 Y=AM*SIN(X/FR+AD)/COS(TA*(X/FR+A
D)):IFCT=1THENY=-Y
210 Y=512+Y:IFY<0THENY=0
220 IFY>1023THENY=1023
230 RETURN
```

Programa 2

```
5 REM 3-D PIRAMIDE
10 GRAPHIC2:COLOR0,7,1,12
20 FORI=1TO5:READX(I),Y(I),Z(I):NEXT
```



```
100 FORI=1TO5
105 YY(I)=Y(I)
110 XX(I)=X(I)
111 ZZ(I)=Z(I)
120 POINT2,500+(200*XX(I))/(YY(I)+20
0),500
130 NEXT:XC=0:YC=220
160 A=A+20:R=A/57.29
165 FORI=1TO5
170 XX(I)=(X(I)-XC)*COS(R)+(Y(I)-YC)
*SIN(R)+XC
180 YY(I)=(Y(I)-YC)*COS(R)+(X(I)-XC)
*SIN(R)+YC+100
185 AX(I)=(300*XX(I))/(YY(I)+300)+50
0:AY(I)=(300*ZZ(I))/(YY(I)+300)+500
190 NEXT
191 SCNCLR
195 FORI=1TO4
200 DRAW2,AX(5),AY(5)TOAX(I),AY(I)
210 NEXT
220 DRAW2,AX(1),AY(1)TOAX(3),AY(3)
221 DRAW2,AX(3),AY(3)TOAX(4),AY(4)
222 DRAW2,AX(2),AY(2)TOAX(1),AY(1)
223 DRAW2,AX(2),AY(2)TOAX(4),AY(4)
990 GOTO160
1000 DATA-200,20,190,-200,420,190,20
0,20,190,200,420,190,0,220,-800
```

Programa 3

```
5 REM GRAPHIC DEMO
6 REM KINETIC STRING ART ADAPTED FRO
M BYTE ARTICLE
8 REM NECESITA AMPLIACION DE MEMORIA
 DE 16K
10 GRAPHIC2
20 COLOR0,0,5,5
30 FORL=0TO500STEP40
40 CIRCLE2,511,511,L,500-L
50 NEXT
55 GOSUB900
60 Q=RND(1)/2+.25
70 FORL=0TO500STEP40
80 CIRCLE2,511,511,L,500-L*Q
90 NEXT
100 GOSUB900
110 Q=RND(1)/2+.25
120 FORL=0TO500STEP40
130 CIRCLE2,511,511,L*Q,500-L
140 NEXT
150 GOSUB900
220 FORL=0TO2*πSTEPπ/25
230 CIRCLE2,511+400*SIN(L),511+400*C
OS(L),75,100
240 NEXT
250 GOSUB900
260 FORL=0TO2*πSTEPπ/50
270 CIRCLE2,511+400*SIN(L),511+400*C
OS(L)/(L+1),75-L*10,100-L*13
280 NEXT
290 GOSUB900
300 GRAPHIC1:DIMA%(200,3):FORL=1TO5:
SC=INT(RND(1)*8)
301 BO=INT(RND(1)*16):IFBO=SCTHEN301
302 CH=INT(RND(1)*8):IF(CH=SC)OR(CH=
BO)THEN302
303 AU=INT(RND(1)*16):IF(AU=SC)OR(AU
=BO)OR(AU=CH)THEN303
304 COLORSC,BO,CH,AU
310 X=INT(RND(1)*1024):Y=INT(RND(1)*
1024):X1=INT(RND(1)*1024)
315 Y1=INT(RND(1)*1024)
320 C1=0:C2=0
330 FORM=0TO200
340 IFC1<1THENC1=5+INT(RND(1)*10):RE
=INT(RND(1)*3)+1
350 IFC2<1THENDX=INT(RND(1)*81)-40:D
Y=INT(RND(1)*81)-40:DA=INT(RND(1)*81
)-40
355 IFC2<1THENDB=INT(RND(1)*81)-40:C
2=15+INT(RND(1)*10)
360 X=X+DX:Y=Y+DY:X1=X1+DA:Y1=Y1+DB
361 IF(X<0)OR(X>1023)THENDX=-DX:X=X+
DX*2
362 IF(Y<0)OR(Y>1023)THENDY=-DY:Y=Y+
DY*2
363 IF(X1<0)OR(X1>1023)THENDA=-DA:X1
=X1+DA*2
364 IF(Y1<0)OR(Y1>1023)THENDB=-DB:Y1
=Y1+DB*2
365 C1=C1-1:C2=C2-1
370 DRAW0,A%(M,0),A%(M,1)TOA%(M,2),A
%(M,3)
380 DRAWRE,X,YTO1,Y1:A%(M,0)=X:A%(M,
1)=Y:A%(M,2)=X1:A%(M,3)=Y1
390 NEXT:NEXT
400 GOSUB900
899 GRAPHIC4:COLOR1,3,6,6:PRINT"ESPE
RO[SPC]QUE[SPC]LE[SPC]HABRA[SPC]GUST
ADO[SPC]EL[SPC]SHOW!":GOSUB900:RUN
900 FORD=1TO2000:NEXT:SCNCLR:RETURN
```



SEGUNDA PARTE

EL CHIP DE VIDEO DEL COMMODORE-64

En este artículo acabaremos de ver los registros del chip de vídeo

Por R. PARDO

REGISTRO 23
(Posición 53271 - $D017)

Es el registro de expansión vertical de los Sprites. El bit 0 provocará la expansión vertical del Sprite 0, el bit 1 hará lo mismo con el Sprite 1 y así sucesivamente. La expansión será efectiva poniendo el correspondiente bit a 1 (0=tamaño normal, 1=tamaño expandido).

REGISTRO 24
(Posición 53272 - $D018)

Es un registro de control de memoria del chip de vídeo. Analizaremos el contenido de este registro a continuación:

Bits 7-4: Es la dirección de base de la matriz de vídeo. La matriz de vídeo es un área de memoria de 1000 bytes consecutivos los cuales contienen un puntero de carácter de 8 bits. Lo diré en otras palabras: se trata de la dirección de base de la **memoria de pantalla** y los punteros de carácter a los cuales me refiero son los códigos POKE que sitúa de manera automática el chip de vídeo cuando escribimos o cuando hacemos PRINT o POKE en la pantalla. Normalmente está situada a partir de la posición 1024 ($0400).

Bits 3-1: Es la dirección de base del área de memoria donde se almacena la forma de los caracteres. Aquí están contenidos los tres bits más significativos de la dirección de base del **generador de caracteres**. Los 8 bits de orden más bajo están formados por el puntero de carácter en la matriz de vídeo, que seleccionan a un carácter determinado, y a un contador de 3 bits raster que selecciona uno de los 8 bits de carácter (hablaremos más a fondo de este tema cuando toquemos los caracteres definidos por el usuario). Los caracteres resultantes se formatean en una pantalla de 25 filas por 40 columnas. Además hay un nyble (medio byte) de COLOR que está asociado con cada puntero en cada posición de la memoria (la memoria de la matriz vídeo debe ser de una longitud de 12 bits) lo cual define uno de los dieciséis colores para cada carácter.

Bit 0: No se utiliza.



REGISTRO 25
(Posición 53273 - $D019)

Este es el registro de estado de interrupciones. Aquí se muestra el estado de las cuatro fuentes de interrupciones. Pero, ¿qué es una interrupción? Si estamos ejecutando una tarea (por ejemplo jugando a marcianos con nuestro ordenador) y a nuestra/o novia/o se le ocurre llamarnos por teléfono para darnos los buenos días, el hecho de esta llamada nos obligará a **interrumpir** nuestra tarea normal. Cuando hayamos acabado con nuestra conversación volveremos a nuestra tarea normal. Este es el concepto de interrupción.

Las causas que pueden provocar una interrupción son las siguientes:

—Cuando el valor del raster sea igual al valor almacenado en el registro de raster (Registro 18).

—Cuando se produce una colisión entre un Sprite y datos en la pantalla (solamente con la primera colisión se activa la interrupción. Hasta que no se desactive el correspondiente bit de habilitación, no atenderá a más requerimientos de interrupción de esta fuente).

—Cuando se produce una colisión entre Sprites (sólo la primera colisión).

—Cuando se detecta un flanco negativo en la entrada de lápiz óptico (una vez por cuadro).

Una vez vistas las posibles fuentes de interrupción en el VIC-II chip, hablaremos ahora a nivel de bit de este registro:

—Bit 7: Este bit se pone a 1 en cuanto se detecta una fuente de interrupción habilitada y activada, provocando la interrupción en el microprocesador.

Bits 6-4: Estos bits no se usan. Normalmente están a 1.

—Bit 3: Este bit se pone a uno cuando detecta un flanco negativo en la entrada de lápiz óptico, esté o no esté seleccionada el correspondiente bit de habilitación de interrupción (como hemos dicho antes, una vez por cuadro).

—Bit 2: Este bit se pone a uno cuando se detecta una colisión entre Sprites pero sólo cuando se detecta por primera vez.

Bit 1: Este bit se pone a uno cuando se detecta una colisión entre un Sprite y datos pero sólo cuando se detecta por primera vez.

Bit 0: Este bit se pone a uno cuando el valor raster actual es igual al valor almacenado en el Registro 18 (registro raster).

REGISTRO 26
(Posición 53274 - $D01A)

Este es el registro de habilitación de interrupciones. Si una determinada fuente de interrupción requiere de éstas y ADEMAS el correspondiente bit de este registro está habilitado (es decir, está a 1) entonces se activa el bit más significativo del registro anterior que es en definitiva el que provoca el requerimiento de interrupción al microprocesador. Veremos este registro a nivel de bit:

Bits 7-4: No se usan. Normalmente están a uno.

Bit 3: Este bit habilita la interrupción desde el lápiz óptico.

Bit 2: Este bit habilita la interrupción de colisión entre Sprites.

Bit 1: Este bit habilita la interrupción de colisión entre Sprite y datos de la pantalla.

Bit 0: Este bit habilita la interrupción desde el registro raster.

REGISTRO 27
(Posición 53275 - $D01B)

Este registro le indica al chip de vídeo la prioridad que tienen los Sprites con



respecto a los datos de la pantalla, es decir, si un determinado Sprite va a pasar por encima o por debajo de los datos en la pantalla. Cada bit en este registro tiene relación con un Sprite determinado: el bit 0 determinará la prioridad del Sprite 0, el bit 1 determinará la prioridad del Sprite 1, y así sucesivamente (1=Sprite encima datos; 0=Sprite debajo datos).

REGISTRO 28
(Posición 53276 - $D01C)

Este registro es el que nos permite seleccionar el modo de color para un determinado Sprite. Hay dos modos de color: **Modo Color Standard** y **Modo Multicolor**. En el Modo de Color Standard un 0 en un bit de patrón de Sprite permite que se vea cualquier información que haya detrás (se transparenta la información) mientras que un 1 en un bit de patrón de Sprite hace que aparezca como un color determinado por el correspondiente Registro de Color de Sprite. En el Modo Multicolor se selecciona este modo para cada Sprite independiente mediante este registro. Cuando el bit correspondiente esté a 1, el Sprite correspondiente se mostrará en Modo Multicolor. En el Modo Multicolor, los datos del patrón del Sprite se interpretan a **pares** (igual que en los otros Modos Multicolor que veremos más adelante) como sigue:

Par de Bits	Color Mostrado
00	Transparente
01	Color de Sprite 0 (Registro 37)
10	Color de Sprite (Registros 39-46)
11	Color de Sprite 1 (Registro 38)

Cada bit de este registro está relacionado con un Sprite determinado, es decir, el bit 0 afecta al Sprite 0, el bit 1 afecta al Sprite 1, y así sucesivamente (1=multicolor, 0=standard).

REGISTRO 29
(Posición 53277 - $D01D)

Este es el registro de expansión horizontal de Sprites. Cada bit tiene relación con un Sprite determinado: el bit 0 actuará sobre el Sprite 0, el bit sobre el Sprite 1, y así sucesivamente (1=expansión, 0=normal).

REGISTRO 30
(Posición 53278 - $D01E)

Este es el registro de detección de colisión entre Sprites. Una colisión entre dos Sprites ocurre cuando las partes no transparentes de dos Sprites coinciden. La coincidencia de áreas transparentes de Sprites no generan la colisión. Cuando ocurre una colisión, los bits correspondientes a los Sprites en colisión de este registro se pondrán a 1. Cuando una colisión entre dos o más Sprites, el bit de colisión Sprite-Sprite para cada Sprite que haya intervenido en esta se pondrá a 1. Los bits de colisión estarán a 1 hasta que se lea este registro tras lo cual se pondrán automáticamente todos los bits a 0. Las colisiones entre Sprites también se detectan si ocurren fuera de la pantalla. Como en otros registros, cada bit informa del estado del Sprite correspondiente (1=colisión, 0=normal).

REGISTRO 31
(Posición 53279 - $D01F)

Este es el registro de detección de colisión entre Sprites y Datos. Este tipo de colisión se puede dar entre Sprites y datos de la pantalla de Baja Resolución o puntos de la pantalla de Alta Resolución. Este registro tiene un bit para cada Sprite el cual se pone a 1 cuando coinciden datos y Sprites. Como en el registro anterior, la coincidencia de las partes transparentes de datos y Sprites no da lugar a una colisión. Para aplicaciones especiales, la presentación del par multicolor 01 tampoco causa una colisión. Esta característica permite usarlo como presentación de fondo de pantalla sin interferir con las verdaderas colisiones de Sprite. Una colisión Sprite-Dato puede ocurrir fuera de la pantalla en la dirección horizontal si los datos de la pantalla se han movido a una posición fuera de pantalla (con la característica de scrolling suave). También este registro se pone a cero inmediatamente después de efectuarse su lectura.

REGISTRO 32
(Posición 53280 - $D020)

Este es el registro que contiene la información del color de marco de la pantalla en los cuatro bits más bajos. Los cuatro bits más significativos de este registro no se usan y están normalmente a 1. Los códigos de color son los siguientes:

0	Negro	8	Naranja
1	Blanco	9	Marrón
2	Rojo	10	Rosa
3	Cian	11	Gris claro
4	Púrpura	12	Gris
5	Verde	13	Verde claro
6	Azul	14	Celeste
7	Amarillo	15	Gris oscuro

REGISTRO 33
(Posición 53281 - $D021)

Este es el registro que controla el color de fondo de pantalla. Los cuatro bits más significativos no se usan y están normalmente a 1. Los cuatro bits menos significativos son los que controlan el color de fondo. En Modo Carácter Multicolor es el color 0.

REGISTRO 34
(Posición 53282 - $D022)

Este registro controla el color de fondo de pantalla 1. Solamente se utiliza en Modo Multicolor.

REGISTRO 35
(Posición 53283 - $D023)

Este registro controla el color de fondo de pantalla 2. Solamente se utiliza en Modo Multicolor.

REGISTRO 36
(Posición 53284 - $D024)

Este registro controla el color de fondo de pantalla 3. Solamente se utiliza en Modo Multicolor.

REGISTRO 37
(Posición 53285 - $D025)

Este registro selecciona el color de Sprite Multicolor 0. Este color es común para todos los Sprites Multicolor. Los cuatro bits menos significativos seleccionan el color y los cuatro bits más significativos no se usan, estando normalmente a 1.

REGISTRO 38
(Posición 53286 - $D026)

Este registro selecciona el color de Sprite Multicolor 1. Este color es común para todos los Sprites Multicolor. Los cuatro bits menos significativos seleccionan el color y los cuatro bits más significativos no se usan, estando normalmente a 1.

REGISTRO 39
(Posición 53287 - $D027)

Este es el registro de color del Sprite 0. Los cuatro bits menos significativos controlan el color del Sprite.

REGISTRO 40
(Posición 53288 - $D028)

Este es el registro de color del Sprite 1. Los cuatro bits menos significativos controlan el color del Sprite.

REGISTRO 41
(Posición 53289 - $D029)

Este es el registro de color del Sprite 2. Los cuatro bits menos significativos controlan el color del Sprite.

REGISTRO 42
(Posición 53290 - $D02A)

Este es el registro de color del Sprite 3. Los cuatro bits menos significativos controlan el color del Sprite.

REGISTRO 43
(Posición 53291 - $D02B)

Este es el registro de color del Sprite 4. Los cuatro bits menos significativos controlan el color del Sprite.

REGISTRO 44
(Posición 53292 - $D02C)

Este es el registro de color del Sprite 5. Los cuatro bits menos significativos controlan el color del Sprite.

REGISTRO 45
(Posición 53293 - $D02D)

Este es el registro de color del Sprite 6. Los cuatro bits menos significativos controlan el color del Sprite.

REGISTRO 46
(Posición 53294 - $D02E)

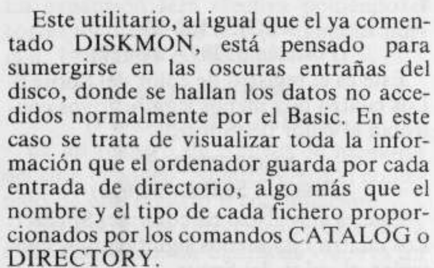
Este es el registro de color del Sprite 7. Los cuatro bits menos significativos controlan el color del Sprite. ■



Rincón del 700

SOFTWARE PARA EL 700 (8) UTILITARIOS (5)





Por Jordi SASTRE

Este utilitario, al igual que el ya comentado DISKMON, está pensado para sumergirse en las oscuras entrañas del disco, donde se hallan los datos no accedidos normalmente por el Basic. En este caso se trata de visualizar toda la información que el ordenador guarda por cada entrada de directorio, algo más que el nombre y el tipo de cada fichero proporcionados por los comandos CATALOG o DIRECTORY.

La información que facilita permite conocer la ubicación física de los ficheros en el disco, dónde están los "side sector" de los ficheros relativos, el anterior emplazamiento de los ficheros recién borrados, su posible recuperación, etc.

El uso de LOOKDIR va muy unido al de DISKMON. En realidad, la información que proporciona LOOKDIR también la puede ofrecer el DISKMON, sólo que LOOKDIR la visualiza bajo un correcto formato puesto que conoce exactamente el significado de cada uno de los bytes que lee.

Su utilidad es esporádica, porque este tipo de acceso al directorio de un disco no es muy frecuente, pero vital en los casos en que se necesita conocer alguno de estos datos.

El utilitario lo componen tres ficheros:

—LOOKDIR Programa principal.

—LOOKDIR.DR Descriptor de pantalla asociada.

—LOOKDIR.FP Pantalla asociada.

Por consiguiente, si alguien desea copiar este utilitario a otro disco, debe copiar los tres ficheros.

El programa se pone en marcha cargando y ejecutando el programa LOOKDIR: DLOAD "LOOKDIR" y RUN.

Lo primero que hace es cargar del disco la pantalla que utilizará para mostrar los directorios. Esta carga la efectúa desde el drive 0 del periférico 8. Si por cualquier causa, esta pantalla estuviera en otro drive u otro periférico, deberá modificarse convenientemente la línea 80 del programa LOOKDIR.

En la parte superior de la pantalla el programa pide el formato (8050, 8250, 9060 ó 9090), el número de periférico (de 8 a 15) y el drive (0 ó 1) del directorio a visualizar. Si se introducen datos correctos, es decir, que el directorio solicitado es accesible, se empiezan a mostrar los siguientes datos:

—CABECERA DISCO: Visualiza el nombre del disco entre comillas y su identificador.

—TIPO DE FICHERO: Hace constar las siglas SEQ, PRG, USR o REL según el fichero sea de tipo Secuencial, Programa, Usuario o Relativo, respectivamente. Si en esta zona aparecen cuatro asteriscos (****) significa que el fichero está borrado (se le efectuó un SCRATCH). Cuando se borra un fichero, el DOS únicamente pone a cero el byte de tipo de fichero y libera los bloques que ocupa el mismo. Si desea recuperarse este fichero, basta con restaurar en este byte el tipo de fichero (puede usarse el utilitario DISKMON) con su valor correspondiente:

129 = SEQ (Fichero Secuencial)
130 = PRG (Fichero Programa)
131 = USR (Fichero Usuario)
132 = REL (Fichero Relativo).

Y volver a ocupar (B-A) los bloques. No hay ninguna garantía de que estos bloques no hayan sido usados por posteriores grabaciones en el disco desde que se borrara este fichero, en cuyo caso este fichero sería irrecuperable. Más adelante comentaré con mayor amplitud el tema de ficheros borrados. Si a continuación del tipo de fichero aparece un asterisco (por ejemplo, SEQ* o PRG*), significa que éste no ha sido cerrado convenientemente, por lo que no es accesible desde Basic (puede intentarse acceder a él con el DISKMON).

—DIRECCION DE INICIO: Track y Sector del primer bloque usado por el fichero. A partir de éste, los demás bloques que componen el mismo están, encadenados mediante los bytes 0 (track) y 1 (sector) de cada uno.

—NOMBRE: Nombre del fichero. En el disco está rellenado con CHR$ (160).

—PRIMER SIDE SECTOR: Si es un fichero relativo, aquí se hace constar el track y el sector del primero de los side-sectors. En caso contrario los dos bytes están a cero.

—BYTES POR RECORD: Longitud de cada record (si se trata de un fichero relativo).

—NO USADOS: Los bytes 25 a 28 de cada entrada de directorio no se usan. De todas maneras el utilitario los visualiza por si en alguna futura versión de DOS llegan a usarse.

—DIRECCION DE REEMPLAZO: Estos bytes los utiliza interna y momentáneamente el DOS para grabar ficheros con reemplazo (@). Sólo aparecen en el directorio cuando una grabación de este tipo ha sido interrumpida.

—BLOCKS OCUPADOS: Contenidos en los bytes 29 (peso bajo) y 30 (peso alto). Indica la ocupación, en bloques de disco del fichero.

Durante la visualización, la tecla C= (logo Commodore) permite detener momentáneamente la impresión (se reanuda pulsando cualquier otra tecla). Si se pulsa F10 el contenido actual de la pantalla es volcado sobre la impresora (no es necesario pulsar C= previamente). Y la tecla ESC interrumpe la visualización para volver al principio del programa.

Para finalizar pulsar ESC cuando el programa pida FORMATO.

Puede ser interesante, ya que lo he mencionado, comentar un poco más profundamente cómo se utilizan los bytes de "dirección de reemplazo" (bytes 27 y 28).



Es sabido que los programas o ficheros secuencias admiten un tipo de grabación llamada "de reemplazo". Se solicita poniendo el carácter "@" inmediatamente antes del drive (en SAVE y OPEN) o del nombre del fichero (en DSAVE y DOPEN). Por ejemplo, DSAVE "@PRUEBA" provoca que se grabe en el disco el programa PRUEBA borrando cualquier otro programa que ya existiera con este nombre.

Esta forma de grabación puede dar a entender que el sistema operativo borra primero el programa del disco antes de grabar el nuevo. Pero no es así. El proceso exacto que sigue es grabar primero el nuevo programa en otra zona de disco, guardando su dirección de inicio en los bytes 27 y 28 ("dirección de reemplazo"). Una vez grabado completamente, procede a liberar los bloques que ocupaba el anterior programa, el primero de los cuales viene indicado en los bytes 2 y 3 ("dirección de inicio"). Por último, toma los bytes de la dirección de reemplazo y los pone en la dirección de inicio, dejando a cero los primeros.

Vemos, pues, que la dirección de reemplazo está siempre a cero excepto cuando se está ejecutando la grabación con reemplazo. El poder visualizarlos tiene su ventaja. Cuando una grabación con reemplazo es interrumpida por cualquier causa, en el directorio aparece el asterisco (*) indicando que el fichero no ha sido cerrado correctamente por lo que queda inaccesible desde Basic. Pero en el directorio aún consta la dirección anterior del fichero y la nueva dirección, por lo que es posible analizar los bloques del antiguo y del nuevo fichero para intentar recuperar el máximo de información posible ¿Cómo recuperar un fichero?

Depende del tipo de fichero de que se trate. Los programas y ficheros secuenciales son relativamente sencillos de recuperar. Los relativos tienen más dificultad a causa de los side-sectors.

Para dar un ejemplo, supongamos un programa, que por error hemos borrado del disco (SCRATCH). Si nos damos cuenta inmediatamente, antes de efectuar ninguna grabación en el disco, tenemos la completa seguridad de recuperarlo íntegramente. Es muy importante que no se hayan efectuado grabaciones después del SCRATCH, pues podrían ser utilizados algunos de los bloques que antes ocupaba el programa, en cuyo caso las posibilidades de una total recuperación se ven seriamente disminuidas.

Cargamos el DISKMON y visualizamos el primer bloque del directorio (39,1 en 8050 y 8250). Si nuestro programa no está en este bloque (son ocho ficheros de directorio por bloque) visualizamos el siguiente, cuyo track y sector nos indican los bytes 0 y 1 de éste que tenemos en pantalla. Analizaremos los bloques del directorio, siguiendo este encadenamiento entre bloques, hasta localizar en alguno de ellos nuestro programa.

Una vez lo tenemos, modificamos el byte de tipo de fichero, que es el primero de la entrada, tres más a la izquierda del inicio el nombre. Lo identificamos porque actualmente tiene el valor 128 (fichero borrado). Lo modificamos y le ponemos 130 (fichero programa: PRG). Pulsamos ESC, grabamos el bloque, etc. y salimos del programa.

LOOK.DIR — MEC - SOFTWARE

FORMATO: 8050
PERIFERICO: 8
DRIVE: 0

CABECERA DISCO: "UTILITARIOS 700 " C7

1 TIPO FICH	2 DIRECC. INICIO	3	4 A 19 NOMBRE	SIDE SECTOR	22 BYTES REG	25 26 27 28 (NO USADOS)	27 28 DIR. REEMPLAZO	29/30 BLOQUES
PRG	38	1	COPYFILE	[illegible]				14
PRG	38	7	COPY.CM	[illegible]				2
PRG	38	8	LOOKDIR	[illegible]				7
SEQ	38	13	LOOKDIR DR	[illegible]				1
SEQ	38	23	LOOKDIR.FP	[illegible]				8
PRG	40	0	DISKMON	[illegible]				11
SEQ	40	1	DISKMON.DR	[illegible]				10
SEQ	40	8	DISKMON.FP	[illegible]				8
PRG	37	0	UNIT CHANGE	[illegible]				4
PRG	37	2	HEADER CHANGE	[illegible]				12
PRG	37	6	FORMAT8000	[illegible]				12

PULSAR <C=> PARA DETENER, <F10> PARA IMPRIMIR, <ESC> PARA INTERRUMPIR.

Ejemplo del Lookdir en funcionamiento (diseño: J. Sastre; Foto: P. Masats).

Ahora el comando CATALOG o DIRECTORY ya visualiza nuestro programa, y por tanto es accesible desde Basic. Lo que hacemos es leerlo (DLOAD) y grabarlo en otro disco, pues este lo tenemos medio alterado y es conveniente utilizarlo sólo para lecturas (no grabar nada en él) hasta que estemos seguros de haber recuperado toda la información que precisamos. Cuando hemos acabado el traspaso del programa (o programas) al otro disco, hemos de utilizar de nuevo el DISKMON para volver a poner el byte de tipo de fichero a 128 y así no tener engañado por más tiempo al ordenador, y poder seguir utilizando este disco normalmente.

Hay otra manera de recuperar el programa, algo más complicada pero más "correcta": volviendo a ocupar en el BAM los bloques que ocupaba el programa.

Se procede de la misma manera ya explicada, pero en lugar de traspasar la información a otro disco, hemos de escribir un programa para volver a reservar en el BAM los bloques que ocupa el fichero.

Antes hemos de averiguar cuáles eran los bloques que ocupaba el programa. Ponemos en marcha el LOOKDIR y solicitamos el directorio del disco. Localizamos nuestro programa y anotamos la dirección de inicio (bytes 2 y 3, track y sector respectivamente).

El programa ha de hacer lo siguiente. A partir de la dirección de inicio localizada con el LOOKDIR, ha de ocupar cada bloque, sabiendo que los bytes 0 y 1 (track y sector respectivamente) apuntan al siguiente bloque. Es decir:

```
10  OPEN 15,8,15: OPEN 2,8,2 "#":
    REM ABRE FICHEROS.
20  INPUT "TRACK Y SECTOR DE
    INICIO"; T, S.
30  PRINT#15, "U1"; 2; 0; T; S: REM
    CARGA BLOQUE.
40  PRINT#15, "B-P"; 2; 0: REM LEE
    TRACK Y SECTOR DEL
50  GET#2, T$, S$: REM SIGUIENTE
    BLOQUE.
60  PRINT#15, "B-A"; 0;T;S: REM
    OCUPA BLOQUE.
70  T=ASC (T$+CHR$(0)): S=ASC(S$
    +CHR$(0))
80  IF T < > 0 THEN 30: REM
    VUELVE AL BUCLE.
90  DCLOSE: REM FIN DE FICHERO.
```





Galería de Soft

Continuación del listado de programas de Microelectrónica y Control

Programas Commodore 64

SISTEMAS OPERATIVOS

FILE/BOSS 10.000
Cartucho de ampliación del Sistema Operativo de la CPU con nuevos comandos para gestión de ficheros relativos, ficheros tablas, empaquetado y desempaquetado de campos, suma y resta en múltiple precisión (22 dígitos) y comandos auxiliares de programación. Desarrollado en el departamento de Software de M.E.C.

CP/M 20.000
Sistema operativo basado en el microprocesador Z-80. Permite que un programa escrito bajo este standard funcione en un Commodore 64. Presentado en cartucho y en disco.

PROGRAMAS DE APLICACIONES

EASY SCRIPT 20.000
Proceso de texto de gran potencia y versatilidad. Versión en disco. Manual en castellano.

EASY CALC RESULT 20.000
Versión simplificada del CALC RESULT. Presentado en cartucho y uso de disco o cinta. Manual en castellano.

MAGIC DESK (Escribe y Archiva) 16.000
Cartucho que en sí reune las opciones proceso de texto y gestión de ficheros de forma simple y eficaz. Tiene abundantes menús de ayuda y se maneja con joystick. Presentado en cartucho y uso de disco. Manual en castellano.

PROGRAMAS EDUCATIVOS

MATEMATICAS I 3.500
En este disco están grabados programas para ayudar al estudio de algunos temas de Matemáticas. En su contenido se incluyen aspectos de:
—Cambio de bases de numeración.
—Teoría de límites.
—Prácticas de división.
—Prácticas de aritmética.
Presentando en disco. Programas en castellano.

HISTORIA I 3.500
Este disco contiene varios programas para facilitar el estudio de diferentes temas de HISTORIA. En los diferentes programas que contiene se analizan acontecimientos sobre:
—Historia Medieval.
—Historia Antigua.
Presentado en disco. Programas en castellano.

GEOGRAFIA I 3.500
Con los programas contenidos en este disco se pueden estudiar mediante el uso de preguntas y respuestas las capitales de los países de Asia y Africa, pudiéndose modificar por el propio educador para adaptarse a otros continentes. Presentado en disco. Programas en castellano.

GEOGRAFIA II 3.500
Este programa permite cargar mapas de diferentes zonas del Mundo, preguntando por la localización de diferentes ciudades. Presentado en disco. Programas en castellano.

JUEGOS EDUCATIVOS 3.500
En este disco están reunidos diversos programas que, en forma de juego, representan una gran ayuda en Educación, desde divertidos juegos para aprender a sumar y a restar, simulaciones de administración económica, prácticas de tiro (aplicación de la ecuación de la parábola) y gestión de un pequeño negocio, permitiendo el uso del ordenador para esclarecer conceptos de difícil comprensión. Presentado en disco. Programas en castellano.

TEMAS MONOGRAFICOS 3.500
En este disco se incluyen programas que explican de forma sencilla el funcionamiento de algunas aplicaciones científicas que utilizamos a diario.
Se incluyen explicaciones sobre:
—El motor de explosión (cuatro tiempos).
—El motor de dos tiempos.
—El motor Diésel.
Presentado en disco. Programas en castellano.

CONOCIMIENTOS GENERALES 3.500
Disco educativo en el que se ponen a prueba los conocimientos del estudiante en campos como Física, Química, Matemáticas, Geografía, Historia. Programas en castellano.

QUIMICA I 3.500
Este disco contiene varios programas para facilitar el estudio de algunos temas de Química. Se incluyen programas sobre:
—Igualación de ecuaciones.
—Formulación de compuestos.
—Problemas de Química.
Programas en castellano.

Programas VIC-20

PROGRAMAS EN CINTA

PROGRAMAS TECNICOS

PROGRAMACION LINEAL 1.000
Método simplex. Cálculo del valor de las variables que satisfaciendo las restricciones hacen máxima o mínima una función. En castellano. Presentado en caja.

MATRICES 1.000
Suma, resta, multiplicación, multiplicación por un escalar e inversión de matrices. En castellano. Presentación en caja.

REGRESIONES I 1.000
Contenido: Regresión lineal: Cálculo por mínimos cuadrados de la recta que se ajusta mejor a una nube de puntos.
Regresión múltiple: Variable dependiente en función de N variables independientes de grado 1. En castellano. Presentado en caja.

REGRESIONES II 1.000
Contenido: Regresión de orden N: Variable dependiente en función de variable independiente de grado N.
Regresión exponencial: Ajuste nube de puntos a una curva exponencial.
Regresión geométrica: Ajuste de una nube de puntos a una curva geométrica.
En castellano. Presentado en caja.

ESTADISTICA I 1.000
Distribución normal, Poisson, binomial, chi-cuadrado, Student, F de Snedecor. En castellano. Presentado en caja.

ESTADISTICA II 1.000
Contenido: Cálculo de la media, varianza y desviación tipo, tanto de la muestra como de la población, estando los datos agrupados o no. Test de chi-cuadrado y test de Student. En castellano. Presentado en caja.

SISTEMAS 1.000
Contenido: Resolución de sistemas de N ecuaciones con N incógnitas.
Resolución de ecuaciones de grado 2 dando las soluciones tanto reales como complejas.
Cálculo de permutaciones y de combinaciones. En castellano. Presentado en caja.

INTEGRACION 1.000
Cálculo de integrales por el método de Gauss.
Cálculo de derivadas de una función en un punto. Interpolaciones tanto lineales como curvilíneas. En castellano. Presentado en caja.

PROGRAMAS APLICACIONES

DIETA 2.000
En función de la edad, sexo, altura, configuración y actividad desempeñada por la persona, da el peso ideal. Pregunta si se desea seguir un régimen, y en función de la actividad física desarrollada da el peso que se debe tener y la pérdida diaria que se debe alcanzar hasta llegar a dicho peso óptimo. Está en castellano y necesita ampliación de 8K. Presentado en caja.

VICALC 1.500
El VIC-20 se comporta como una calculadora y utiliza la notación polaca inversa. Manual de instrucciones en castellano. Presentado en caja.

QUIZ MASTER 2.000
Programa para preparación de cuestionarios. Especialmente adecuado para escuelas. En castellano. Necesita ampliación de 8 K. Presentado en caja.

MANTENIMIENTO DE FICHEROS 3.000
Programa para la creación y actualización de ficheros en cinta. Es configurable por el usuario. Se complementa con el C-221 Listado de Ficheros. En castellano. Necesita ampliación de 3K. Presentado en caja.

LISTADO DE FICHEROS 3.000
Este programa permite listar por impresora los ficheros generados con el programa Mantenimiento de Ficheros, C-220. Es reconfigurable por el usuario. En castellano. No necesita ampliación. Presentado en caja.

ORDENAMIENTO DE FICHEROS 3.000
Programa para el ordenaamiento de ficheros en cinta. Es reconfigurable por el usuario. Se complementa con el C-220 Mantenimiento de ficheros y con el C-221 Listado de ficheros. Necesita ampliación de 3K. Presentado en caja.

PROGRAMAS DE JUEGOS

CODEMAKER 1.500
El VIC-20 inventa un código secreto, teniendo que descubrirlo. Instrucciones en castellano y el programa en inglés. No necesita ampliación.

COMMENTARIOS / COMMODORE WORLD

Commodore 64 Fun and Games

Autores: Jeffries, Fisher, Sawyer.
Precio: 2.390 ptas.

Editorial: Osborne/Mc. Graw-Hill
Sta. Beatriz, 4
28038 Madrid
Teléf.: 433 87 77

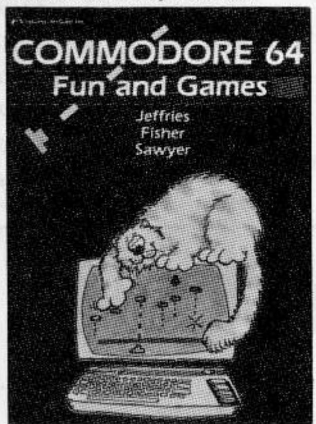
Son 35 programas escritos para el Commodore 64, los programas no sirven para el VIC-20 ni para los PET CBM, ya que utilizan características especiales del C-64 (color, sonido, sprites, etc.). Todos los programas están escritos en BASIC, y no hay que ser un programador profesional para hacer funcionar los juegos que aparecen, sólo hay que teclearlos, ya que son programas que han sido comprobados por varias personas antes de entregarlos en la editorial, y posteriormente el editor los comprobó personalmente.

Nosotros hemos probado un par de juegos y funcionaron bien, el único problema es que vienen en Inglés, y si queréis podéis cambiar los rótulos de los mensajes que aparecen en la pantalla por sus equivalentes en español.

Nos parece útil, que para facilitar la edición de los programas, utilizan un conjunto de rutinas estandar para todos ellos (standard framework) con lo que se gana bastante tiempo una vez tecleado el primer programa.

También vienen las señas donde se pueden pedir los juegos en un disco para ahorrar trabajo.

No podemos decir que los juegos sean una maravilla, rápidos y todo eso, ya que están escritos en Basic y no en código máquina, pero hay que decir que son entretenidos, funcionan y sobre todo son muy didácticos.



Cada juego viene acompañado de una descripción y explicaciones sobre las modificaciones que puedes realizar (si deseas cambiar la puntuación, etc.).

Getting the Most From Your Vic-20

Precio 2.590 ptas.

Hank Librach
Micro text publications, Inc.

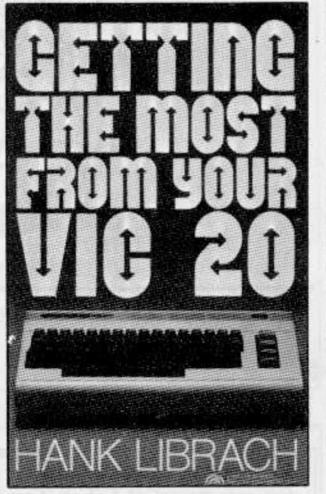
Distribuido por: RA-MA Libros
Chiquinquira, 28 (entr. Cocuy)

Tel. 764 50 95
28033 - Madrid

Este libro está dividido en dos partes, la primera de ellas es una exposición de los comandos Basic más importantes y como se puede aprovechar toda su potencia, la segunda parte está dedicada a programas de aplicación específicos.

Pese a que el libro está en Inglés, se puede entender fácilmente sin tener mucha idea de este idioma, ya que vienen muchos ejemplos y listados.

La primera parte trata detenidamente los puntos que el manual no explica, por ejemplo cómo realizar programas que utilizan el Joystick, cómo crear caracteres programables en pantalla, cómo utilizar las teclas de función, cómo utilizar de modo efectivo, periféricos como el datassette para almacenamiento de datos y programas, etc. Podemos encontrar, en esta parte, infinidad de detalles curiosos de las distintas posibilidades del VIC, tal como encolumnado de números, partes decimales, redondeo de cifras, manejo de cadenas (strings), ordenación, utilización del reloj interno como reloj de tiempo real o como cronómetro en juegos, números aleatorios, música, pantallas, etc. En la segunda parte encontramos diez programas de distintos tipos, cálculo de amortización, tipo de registro de cheques, entrenador lógico, conversión decimal/hexadecimal, entrenador matemático, etc. Al final aparece un conjunto de tablas de utilidad.



COMPARTIENDO EXPERIENCIAS ENTRE AMIGOS



***E**sta sección está dedicada a la colaboración de todos nuestros lectores y está dividida en dos partes:*

1) Programación: Programas y similares.

2) Magia: Trucos, sugerencias, etcétera (pág. 50).

Habrá premios y alicientes para todos los participantes (ver pág. 5).

Enviarnos vuestra dirección para que podáis poneros en contacto unos con otros. ¡Animaros, chicos (...y chicas)!

BINGO

Rafael Muñoz Alfonso
C/Costa Brava, nº 5 (Urb. Can Batlle)
Vallirana (Barcelona)
Teléf.: 660 18 25

C-64

Os mando este programa para el C-64 que os permitirá jugar al BINGO con vuestros amigos y familiares sin necesidad de estar continuamente dando vueltas al bombo, y buscando el sitio donde colocar la bola que ha salido.

No sé si ya habréis recibido algún programa basado en el conocido juego del BINGO, ya que sólo hace dos meses que adquirí un C-64 y no tengo mucha información. Por si no habéis recibido ninguno aquí tenéis uno.

El programa se basa en el conocido juego del bingo y os lo detallo más adelante, sin embargo, me gustaría remarcar que en el momento de salir un valor, la probabilidad que existe de que eso suceda es 1/N, donde N es el número de valores que aún no han salido. Así pues, la probabilidad varía desde 1/90 (al inicio del juego) hasta 1 (a este límite no se llega ya que el bingo debe ser "cantado" por alguien, mucho antes). De esta forma, la probabilidad varía de la misma manera a como si estuviésemos jugando con un bombo y bolas, las cuales vamos extrayendo de una en una. Esta variación de la probabilidad se consigue efectuando una trasposición de valores en el vector V(I) (que contiene todos los valores del bingo) después de que un cierto valor ha sido "cantado", eliminándolo del vector V(I).

La ejecución del programa se inicia con la presentación en pantalla del título del programa, a continuación se pregunta el número de cartones vendidos, el precio de cada cartón, el porcentaje de premio para la línea y el porcentaje para el bingo. Finalizada la entrada de datos aparecen en la pantalla los premios que corresponden a la línea y al bingo.

El programa continúa dibujando un tablero, donde se irán colocando los números (a medida que son "cantados") en su lugar correspondiente. También se dibuja una urna, donde una serie de bolas están animadas de movimiento de ascenso y descenso en el interior de la urna. En un momento determinado, una bola desciende por un canal que comunica la urna con la bandeja, apareciendo el número correspondiente a la jugada en cuestión para ser cantado por el animador del juego. Si ejecutáis vosotros mismos el programa comprobaréis rápidamente cómo funciona por lo que no continúo con esta explicación y paso a detallaros un poco el programa.

Líneas 1-10: Cambian el color de la pantalla, se inicializa una serie de variables y se llama a las subrutinas que presenta el título del programa y la entrada de datos necesarios.

Líneas 20-100: Se dibuja la urna, el tablero y las bolas en la urna. También se inicializa el vector V(I).

Línea 110: Se calcula un número para el bingo a partir del índice del vector V(I).

Líneas 130-180: Se escogen tres bolas de la urna para que se muevan en el interior de ella.

Líneas 190-200: Se controla el movimiento ascendente y descendente de las bolas.

Líneas 217-290: Se presenta en pantalla el número que ha salido en el bingo y se envía la ejecución al cálculo de un nuevo número para la siguiente jugada.

Líneas 300-380: Corresponden a la subrutina que provoca el movimiento de las bolas y controla el sonido de las mismas.

Líneas 400-430: Esta subrutina calcula la posición del número que ha sido "cantado" en el tablero.

Líneas 440-470: Se efectúa una trasposición de valores en el vector V(I) para que la probabilidad varíe entre los límites ya indicados.

Líneas 475-620: Controlan la tecla pulsada cuando se "canta" línea o bingo, y se imprime en pantalla un mensaje consecuente.

Líneas 630-670: Corresponden a la escritura de mensajes de final de juego.

Líneas 680-790: Esta subrutina pide los datos necesarios al inicio del juego.

Líneas 800-810: Controlan si se ha pulsado una tecla o no.

Líneas 820-830: Activan las notas musicales.

Líneas 880-1000: Esta subrutina presenta el título del programa al inicio del juego.

Líneas 1130-1140: Posicionan el cursor en el lugar donde se imprimen los distintos mensajes.

Líneas 1150-1160: Borran los mensajes que aparecen en la pantalla a lo largo del juego.

Líneas 1170-2000: Envían la ejecución a las dos subrutinas anteriores (líneas 1130-1160).

BINGO

```
1 POKE53280,5:POKE53281,5
2 A=1305:B=1[illegible]:C=54272:W=1032:E=1184
3 F=1824:G=1960:DIMV(90)
4 PRINT"[CLR][19CRSRD][11CRSRR][BLK]
ESPERE[SPC]POR[SPC]FAVOR":FORT=1TO300
0:NEXTT
5 GOSUB880
10 GOSUB680
20 LL=0:BB=0:P=0:PRINT"[CLR][BLK]"
30 FORI=W-31TOW:POKEI,121:POKEI+320,1
20:NEXTI
40 FORT=W+32TOA-1STEP40:POKET,117:PO
KET+8,118:NEXTT
50 FORI=E-1TOF-41STEP40:POKEI,118:PO
KEI+2,117:NEXTI
60 FORT=F+39TOF+41:POKET,120:NEXTT
70 FORI=1TO9:FORT=1TO10:W=W+3:POKEW,
102:POKEW+1,102:NEXTT:W=W+50:NEXTI
80 FORI=1TO90:V(I)=I:NEXTI:MAX=90:TT
=1
85 PRINT"[22CRSRD]NUMERO[SPC]ANTERIO
R"
87 PRINTTAB(19)"[2CRSRU]PULSAR[SPC][RVSON]
L[RVSOFF][SPC]PARA[SPC]LINEA"
88 PRINTTAB(19)"PULSAR[SPC][RVSON]B[RVSOFF]
[SPC]PARA[SPC]BINGO[4CRSRU]"
90 FORI=ATOB:POKEI+C,6:POKEI,81:NEXT
I
95 POKEC+24,15:POKEC+6,128:POKEC+5,1
28
100 IFTT>90THEN630
110 NU=INT(RND(TI)*MAX)+1:W=1032:GOS
UB400
130 FORK=1TO2
140 X1=INT(RND(TI)*(B-A+1))+A
150 X2=INT(RND(TI+1)*(B-A+1))+A
160 IFX2=X1THEN150
170 X3=INT(RND(TI+2)*(B-A+1))+A
180 IFX3=X1ORX3=X2THEN170
190 L=-1:GOSUB300
200 L=1:X1=D-40:GOSUB300:GOSUB800:IF
P=1THENK=2
210 NEXTK
215 IFP=1THEN630
217 POKEC+4,33
220 FORI=A+3TO1788STEP40:POKEI+C,5:P
OKEI,32:POKEI+40+C,6:POKEI+40,81
225 POKEC+1,I/50:POKEC,I/200AND255:N
EXTI:POKEC+4,0
230 GOSUB800:IFP=1THEN630
240 POKEF+C,7:POKEF,FI+48:POKEF+1+C,
7:POKEF+1,CO+48:FORT=1TO300:NEXTT
250 POKEPO+C,1:POKEPO,FI+48:POKEPO+1
+C,1:POKEPO+1,CO+48
260 POKEG+C,1:POKEG,FI+48:POKEG+1+C,
1:POKEG+1,CO+48
270 GOSUB800:IFP=1THEN630
280 POKEF+C,5:POKEF,32:POKEF+1+C,5:P
OKEF+1,32:POKEE+C,0:POKEE,120
283 POKEA+3+C,6:POKEA+3,81
285 GOSUB800:IFP=1THEN630
290 GOSUB440:GOTO100
300 POKEC+4,17:I2=0:I3=0
310 FORI=1TO5
320 IFI>2THENI2=40
330 IFI>3THENI3=40
340 D=X1+40*L*(I-1)
350 POKED+C,5:POKED,32:POKEX2+C,5:PO
KEX2,32:POKEX3+C,5:POKEX3,32
360 X2=X2+I2*L:X3=X3+I3*L
365 NT=50+L*(12*I-42):POKEC+1,NT:POK
EC,NT/2AND255
370 POKEX2+C,6:POKEX2,81:POKEX3+C,6:
POKEX3,81:POKED+C+40*L,6:POKED+40*L,
81
380 NEXTI:POKEC+4,0:RETURN
400 FI=INT(V(NU)/10):CO=V(NU)-FI*10:
PO=W+80*FI
410 IFCO<>0THENPO=PO+CO*3:GOTO430
420 PO=PO-50
430 RETURN
440 IFNU=MAXTHEN460
450 FORT=NUTOMAX-1:V(T)=V(T+1):NEXTT

460 MAX=MAX-1:TT=TT+1
470 RETURN
475 GOSUB1130
480 IFLL=1ANDS$="L"THENPRINT"LINEA[SPC]
YA[SPC]CANTADA":GOTO550
490 IFBB=1ANDS$="B"THENPRINT"BINGO[SPC]
YA[SPC]CANTADO":GOTO550
500 IFS$="L"THENLL=1:PRINT"LINEA[SPC]
CORRECTA[SPC](S/N)?":GOSUB560:GOTO57
0
510 IFS$="B"THENBB=1:PRINT"BINGO[SPC]
CORRECTO[SPC](S/N)?":GOSUB560:GOTO59
0
520 PRINT"ERROR[SPC]AL[SPC]PULSAR":P
RINTTAB(17)"PULSE[SPC]DE[SPC]NUEVO[SPC]
[RVSON]L[RVSOFF][SPC]O[SPC][RVSON]B[RVSOFF]
"
525 GETS$:IFS$=""THEN525
530 GOSUB1170
540 GOTO 480
```



¿Que no sabes cómo suscribirte a Commodore World?

¡¡¡PUES VENGA, LLAMANOS!!!
(91) 231 23 85/86 y (93) 212 73 45

```
550 PRINTTAB(17)"CONTINUAMOS":FORK=1
TO3000:NEXTK:GOSUB1170:GOTO620
560 GETS$:IFS$=""THEN560
565 IFS$<>"S"ANDS$<>"N"THEN560
567 RETURN
570 IFS$="S"THENGOSUB1170:PRINT"LINE
A[SPC]CORRECTA":PRINTTAB(17)"GANA"GL
"PTS":GOTO550
580 IFS$="N"THENLL=0:GOSUB1170:PRINT
"LINEA[SPC]NO[SPC]CORRECTA":GOTO550
590 IFS$="N"THENBB=0:GOSUB1170:PRINT
"BINGO[SPC]NO[SPC]CORRECTO":GOTO550
600 IFS$="S"THENGOSUB1170:PRINT"BING
O[SPC]CORRECTO":PRINTTAB(17)"GANA"GB
"PTS":P=1
610 PRINTTAB(17)"FELICIDADES":RESTOR
E:GOSUB820:FORT=1TO400:NEXTT
620 RETURN
630 PRINT"[CLR][CRSRD]QUIERES[SPC]JU
GAR[SPC]OTRA[SPC]VEZ[SPC](S/N)?"
640 GETS$:IFS$=""THEN640
650 IFS$="S"THENPRINT"[CRSRD]ES[SPC]
LO[SPC]MEJOR[SPC]QUE[SPC]PUEDES[SPC]
HACER":FORT=1TO3000:NEXTT:GOTO10
660 IFS$<>"N"THEN630
670 PRINT"[CRSRD]TU[SPC]TE[SPC]LO[SPC]
PIERDES,[SPC]ADIOS":END
680 PRINT"[CLR]"TAB(18)"[BLK][RVSON]
BINGO[RVSOFF]":PRINT"[2CRSRD]":INPUT
"CARTONES[SPC]VENDIDOS";AA:PRINT
690 INPUT"PRECIO[SPC]DE[SPC]CADA[SPC]
CARTON";PP:PRINT
700 INPUT"%[SPC]PARA[SPC]LINEA";SS:P
RINT
710 INPUT"%[SPC]PARA[SPC]BINGO";CC:P
RINT
720 IFSS+CC>100THEN700
730 IFSS+CC<100THENGL=INT(AA*PP*SS/1
00):GB=INT(AA*PP*CC/100):GOTO750
740 GL=INT(AA*PP*SS/100):GB=AA*PP-GL

750 PRINT"[2CRSRD]PREMIO[SPC]A[SPC]L
A[SPC]LINEA"GL"PTS"
760 PRINT"[2CRSRD]PREMIO[2SPC]AL[2SPC]
BINGO"GB"PTS"
770 PRINT"[2CRSRD]PULSE[SPC]UNA[SPC]
TECLA[SPC]PARA[SPC]EMPEZAR"
780 GETS$:IFS$=""THEN780
790 RETURN
800 GETS$:IFS$<>""THENGOSUB475
```

```
810 RETURN
820 POKEC+24,15:POKEC+6,0:POKEC+5,12
8:POKEC+3,99:Z=3000
822 READM1:IFM1=0THENPOKEC+4,0:RETUR
N
824 R=INT(M1/10):Q=(M1-10*R)*Z
826 POKEC+1,Q/256:POKEC,QAND255
828 POKEC+4,0:POKEC+4,65:FORI=1TO40*
(R+1):NEXTI
830 GOTO822
832 DATA26,6,25,5,2,2,2,5,6,7,26,6,2
5,5,2,2,2,64,2,2,5,4,4,5,2,2,4,5,6,7
834 DATA15,6,24,5,2,2,2,5,6,7,26,6,2
5,5,2,2,2,94,0
880 PRINT"[CLR]":H=1027:HH=1427
890 FORI=HTOHHSTEP40:POKEI,42:POKEI+
8,42:POKEI+11,42:POKEI+16,42:NEXTI
900 FORI=H+59TOHH-21STEP40:POKEI,42:
POKEI+8,42:POKEI+13,42:NEXTI
910 FORI=H+1TOH+3:POKEI,42:POKEI+200
,42:POKEI+400,42:NEXTI
920 POKEH+4,42:POKEH+164,42:POKEH+24
4,42:POKEHH+4,42
930 FORI=H+45TOH+125STEP40:POKEI,42:
POKEI+240,42:NEXTI
940 FORI=H+94TOH+334STEP40:POKEI,42:
NEXTI
950 FORI=H+20TOH+23:POKEI,42:POKEI+8
,42:POKEI+400,42:POKEI+408,42:NEXTI
960 POKEH+12,42:POKEH+53,42:POKEH+37
5,42
970 FORI=H+302TOH+304:POKEI,42:NEXTI

980 POKEH+64,42:POKEH+344,42:POKEH+3
84,42
990 PRINT"[16CRSRD][6CRSRR]POR[SPC]R
AFAEL[SPC]MUNYOZ...":FORI=1TO8000:NE
XTI
1000 RETURN
1130 POKE781,18:POKE782,17:POKE783,0
:SYS65520
1140 RETURN
1150 PRINT"[22SPC]":PRINTTAB(17)"[22SPC]
"
1155 PRINTTAB(17)"[22SPC]"
1160 RETURN
1170 GOSUB1130
1180 GOSUB1150
1190 GOSUB1130
2000 RETURN
```

GLOSARIO

Alfanumérico.—Denominación para datos que pueden incluir ambos tipos de caracteres (alfabéticos y numéricos). Por lo general se usa para definir a todos los caracteres imprimibles, incluyendo espacios y símbolos.

ASCII.—(American Standard Code for Information Interchange) Es el método más usado para codificar caracteres en códigos compuestos en grupos de bits. Prácticamente todos los ordenadores personales usan ASCII, así como las grandes computadoras (excepto IBM que usa un código llamado EBCDIC).

El código ASCII está compuesto de 32 caracteres de control que no tienen una imagen imprimible y 96 caracteres imprimibles a los cuales normalmente se les denomina como caracteres gráficos.

BCD.—Una abreviatura para decimales en código binario, un método de representar números decimales asignándoles grupos de cuatro bits binarios a cada dígito de los 10 decimales.

CRT.—(Tubo de Rayos Catódicos) Es una abreviatura utilizada para denominar al tubo de la televisión usado para producir imágenes o caracteres.

K.—(Kilobyte) Una medida del tamaño de la memoria. La mayoría de los ordenadores utilizan 16K como mínimo pero por lo general se considera que hacen falta 48K para los trabajos serios.

HALTEROFILIA

Salvador Tudela Edo.
C/Gran Vía, Carlos III nº 130 "A"
Barcelona-34
Teléf.: 204 18 23

"HALTEROFILIA" es un programa para el VIC-20 sin ampliar, el cual consiste en pulsar la tecla que se indica, el mayor número de veces posible en un periodo de tiempo delimitado por dos "pitidos".

En primer lugar tendrás que levantar 25 kilos, que en el caso de que lo consigas pasarás a elegir el número de kilos tú mismo (lo máximo que he conseguido ha sido levantar 91 kilos).

El funcionamiento del programa es bastante sencillo, únicamente os diré que las líneas 70 a 83, son las que generan el sonido de los dos pitidos, y cuentan las pulsaciones realizadas al igual que el tiempo; mientras que las líneas 3000 a 5050 son las que realizan el dibujo del muñeco levantando las pesas.

VIC-20
Sin ampliación

No quiero despedirme sin felicitaros por la revista, la cual considero que nos es de una gran ayuda para poder comprender mejor el funcionamiento de "nuestro VIC". ■

```
10 POKE52,24:POKE56,24:CLR
20 POKE36869,254
22 FORI=6144TO6668:POKEI,PEEK(26624+
I):NEXTI
23 PRINT"[CLR][4CRSRD][10CRSRR][RVSON]
@A"
25 PRINT"[10CRSRR][RVSON]VW"
26 GOSUB3000
40 PRINT"[3CRSRD][3CRSRR]**HALTEROFI
LIA**"
41 PRINT"[4CRSRD][6CRSRR](19/V/84)"
42 FORT=1TO4000:NEXTT:POKE36879,11
43 PRINT"[CLR]":POKE36869,240
45 PRINT"CUANDO[SPC]OIGAS[SPC]EL[SPC]
PRIMER":PRINT"PITIDO,PULSA[SPC]LA[SPC]
TECLA"
46 PRINT"[CRSRD][3CRSRR]/CRSR(DERECH
A)/":PRINT:L=25
47 PRINT"EL[SPC]MAYOR[SPC]NUMERO[SPC]
DE":PRINT"[CRSRD]VECES[SPC]POSIBLE":
PRINT"[CRSRD]HASTA[SPC]QUE"
48 PRINT:PRINT"SUENE[SPC]EL[SPC]2[SPC]
PITIDO."
49 PRINT"[2CRSRD][3CRSRR][RVSON]PULS
A[SPC]UNA[SPC]TECLA[RVSOFF]"
50 GETX$:IFX$=""THEN50
58 POKE36869,240:POKE36879,145
59 PRINT"[CLR]":PRINT"[3CRSRD][CRSRR]
INTENTA[SPC]LEVANTAR";L:PRINT"[6CRSRR]
KILOS"
60 PRINT"[7CRSRD][6CRSRR]ATENCION!!"

65 FORT=1TO4000:NEXTT
67 GETS$:IFS$<>""THEN67
70 POKE36878,15:POKE36876,215:PRINT"
[3CRSRD][6CRSRR]!!!YA!!!"
75 FORT=1TO500:NEXTT:POKE36878,0
77 H=10:TI$="000000"
78 GETX$:IFX$="[CRSRR]"THENP=P+1
79 IFTI/60>HTHEN82
80 GOTO78
82 PRINT"[2CRSRD][5CRSRR]!!ALTO!!":P
OKE36878,15:POKE36876,215:POKE36879,
28
83 FORT=1TO500:NEXTT:POKE36878,0
84 IFL>PTHEN1000
85 POKE36878,15:FORL=130TO254:POKE36
876,L
86 FORM=1TO40:NEXTM:NEXTL:POKE36879,
143
87 PRINT"[CLR]":PRINT"[8CRSRD]TE[SPC]
HA[SPC]COSTADO[SPC]PERO..."
100 POKE52,24:POKE56,24:CLR:POKE3686
9,254
110 FORI=6144TO6668:POKEI,PEEK(26624
+I):NEXTI
115 PRINT"[CLR][4CRSRD][10CRSRR][RVSON]
@A"
117 PRINT"[10CRSRR][RVSON]VW"
120 GOSUB5000
140 PRINT"[3CRSRD][CRSRR]!!LO[SPC]CO
NSEGUISTE!!"
149 FORG=1TO4000:NEXTG:POKE36869,240
:POKE36879,63
150 PRINT"[CLR]":PRINT"[3CRSRD]CUANT
OS[SPC]KILOS[SPC]QUIERES":INPUT"[CRSRD]
AHORA";L
152 POKE36879,57:GOTO58
1000 PRINT"[CLR]":PRINT"[3CRSRD][4CRSRR]
LO[SPC]SIENTO,[SPC]SOLO"
1010 PRINT"[CRSRD]TENIAS[SPC]FUERZA[SPC]
PARA":PRINT"[CRSRD]LEVANTAR";P:PRINT
"KILOS"
1030 INPUT"OTRO[SPC]JUEGO(S/N)";Y$
1040 IFY$="S"THENGOTO42
1050 STOP
3000 FORI=7168TO7183:READX:POKEI,X:N
EXTI
3030 FORI=7344TO7359:READX:POKEI,X:N
EXTI
3060 RETURN
5000 FORI=7168TO7183:READX:POKEI,X:N
EXTI
5010 DATA48,51,63,59,57,15,15,7
5020 DATA12,204,252,220,156,240,240,
224
5030 FORI=7344TO7359:READX:POKEI,X:N
EXTI
5040 DATA7,7,7,7,3,3,3,7
5050 DATA224,224,224,224,192,192,192
,224
5060 RETURN
```

COMMODORE 64 LE DA ACCESO A MUCHOS ACCESORIOS

Unidad simple de disco (Monofloppy) 170 K.

Cassette.



Plotter e impresora, 4 colores, 14 c.p.s.

Impresora matricial, tractor, 30 c.p.s.

Interface RS232.

Joy Stick.

Paddle.

Cursos de Introducción al BASIC.

COMMODORE 64 LE MUESTRA PARTE DE SUS PROGRAMAS

Utilitarios y lenguajes

MONITOR LENGUAJE MAQUINA.
FORTH.
LOGO.
PILOT.
MACRO ASSEMBLER.
PROGRAMMER'S UTILITIES.
TURTLE GRAPHICS II.
MASTER.

Sistemas operativos

FILE/BOSS.
CP/M.

Programas de aplicaciones

EASY SCRIPT.
Proceso de texto de gran potencia.

CALC RESULT.
Hoja electrónica de cálculo.

EASY CALC RESULT.
Versión simplificada del CALC RESULT.

MAGIC DESK.
Proceso de texto y gestión de ficheros.

AGENDA TELEFONICA.

Programas educativos

MUSIC MACHINE.
MUSIC COMPOSER.
VISIBLE SOLAR SYSTEM.
SPEED/BINGO MATH.
FISICA I.
MATEMATICAS I.
HISTORIA I.
GEOGRAFIA I.
GEOGRAFIA II.
JUEGOS EDUCATIVOS.
TEMAS MONOGRAFICOS.
CONOCIMIENTOS GENERALES.
QUIMICA I.

Juegos

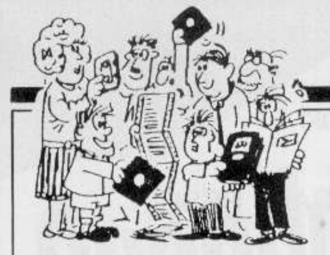
JUPITER LANDER.
KICKMAN.
SEAWOLF.
RADAR RAT RACE.
TOOTH INVADERS.
LAZARIAN.
OMEGA RACE.
LE MANS.
PINBALL SPECTACULAR.
AVENGER.
SUPERMASH.
FROGMASTER.
GRID RUNNER.
ATTACK OF THE MUTANT CAMELS.
THE PIT.
MR. TNT.
6 GAME PROGRAMS.
BINGO.
ROOTING TOOTING.
MINESSOTA FAT'S POOL CHALLENGE.

... y seguimos ampliando la lista

El ordenador personal de la familia más potente

commodore
COMPUTER

MICROELECTRONICA Y CONTROL, S.A.
c/. Taquígrafo Serra, 7, 5.°. Barcelona-29
c./ Princesa, 47, 3.° G. Madrid-8



EL PISTOLERO

Emilio Rocafort Riaza
Urb. Torres San Lamberto, 17-A
Zaragoza

VIC-20
SIN AMPLIACION

El programa **"El pistolero"** es para 2 jugadores. Misión: Cargarte al otro lo más rápidamente posible. Sólo tienes 180 segundos. Hay cinco carromatos que pasan constantemente. También hay cactus. Va para el primer jugador, con shift y commodore y ctrl, y para el segundo con las teclas cursor y con la flecha hacia arriba.

Explicación:

Línea 1-10 Datas

Línea 11 Carga caracteres

Línea 13-15 Coloca cactus e inicialización.

Línea 30-38 Coloca datos y hace movimiento de carretas y de hombres.

Línea 39-85 Comprueba tecla pulsada.

Línea 110 Control del tiempo.

Línea 120 Comprueba balas.

Línea 130 Salto de continuación.

Línea 140-157 Printa balas.

Línea 160-215 Subrutina de tiros.

Línea Fin de juego.

```
2 GOSUB320
4 DATA60,66,36,24,31,24,36,66
5 DATA60,66,36,24,248,24,36,66
6 DATA129,165,165,165,255,24,24,0
7 DATA24,60,102,102,231,195,195,195
8 DATA195,195,255,255,66,66,66,66
9 DATA0,0,0,24,24,0,0,0
10 DATA24,24,24,24,24,24,60,60
11 POKE52,28:POKE56,28:CLR:FORI=7168
TO7168+7*8-1:READX:POKEI,X:NEXT:PRIN
T"[CLR]"CHR$(8)
13 POKE36869,255:TI$="000000"
15 FORI=1TO12:C=INT(RND(1)*18+INT(R
ND(1)*21)*22:POKE7726+C,2:POKE38444+
C,0:NEXT
20 J1=8164:J2=8185:B1=6:B2=6:CP=3072
0:D=160:S=22:C1=0:C2=0:FORI=1TO5:C(I
)=7822+66*I:NEXT
30 PRINT"[HOM][RVSON]MUERTOS"M1"[HOM]
[12CRSRR]MUERTOS"M2
31 PRINT"[HOM][RVSON][CRSRD][9CRSRR]
"VAL(MID$(TI$,3,2))*60+VAL(RIGHT$(TI
$,2)):GOSUB140
35 POKEJ1,0:POKEJ2,1:POKEJ1+CP,0:POK
EJ2+CP,0:POKEJ1+S,0:POKEJ1-S,0:POKEJ
2+S,0:POKEJ2-S,0
36 FORI=1TO5:POKEC(I),4:POKEC(I)-S,3
:POKECP+C(I),0:POKECP+C(I)-S,0:POKEC
(I)+S,0
37 C(I)=C(I)-22:IFC(I)<7734THENPOKEC
(I),0:POKEC(I)+22,0:C(I)=8174
38 NEXTI
39 K1=PEEK(653):K2=PEEK(197)
40 IFK1=1ANDC1=0THENC1=1:IFB1<0THENB
1=0
41 IFK2=31ANDC2=0THENC2=1:IFB2<0THEN
B2=0
50 IFK1=2THENJ1=J1+22:IFJ1>8186THENJ
1=8164
55 IFK1=4THENJ1=J1-22:IFJ1<7724THENJ
1=7724
60 IFK2=23THENJ2=J2+22:IFJ2>8207THEN
J2=8185
65 IFK2=54THENJ2=J2-22:IFJ2<7745THEN
J2=7745
80 IFC1<>0ANDB1<>0THENGOSUB160
85 IFC2<>0ANDB2<>0THENGOSUB190
110 IFTI$>="000300"THEN300
120 IFB1=0ANDB2=0THENPRINT"[CLR]":GO
TO15
130 GOTO30
140 PRINT"[HOM][CRSRD]":IFB1=0THENP
RINT"[RVSON][6SPC]":C1=0:GOTO150
145 FORI=1TOB1:PRINT"F";:NEXTI:FORI=
1TO6-B1:PRINT"[RVSON][SPC][RVSOFF]"
:NEXTI
150 PRINT"[HOM][CRSRD][16CRSRR]":IF
B2=0THENPRINT"[RVSON][6SPC]":C2=0:RE
TURN
155 IF 6-B2=0THEN157
156 FORI=1TO6-B2:PRINT"[RVSON][SPC][RVSOFF]
";:NEXTI
157 FORI=1TOB2:PRINT"F";:NEXTI:RETURN
160 B1=B1-1:D1=J1+1:FORC1=1TO20:A1=D1+C1
165 IFPEEK(A1)=1THENM1=M1+1
170 IFPEEK(A1)=2ORPEEK(A1)=3ORPEEK(A
1)=4THENPOKEA1-1,0:GOTO185
175 POKEA1,5:POKECP+A1,0:POKEA1-1,0
180 NEXTC1
185 C1=0:POKEA1,0:RETURN
190 B2=B2-1:D2=J2-1:FORC2=1TO20:A2=D2-C2
195 IFPEEK(A2)=0THENM2=M2+1
200 IFPEEK(A2)=2ORPEEK(A2)=3ORPEEK(A
2)=4THENPOKEA2+1,0:GOTO215
205 POKEA2,5:POKECP+A2,0:POKEA2+1,0
210 NEXTC2
215 POKEA2,0:C2=0:RETURN
300 PRINT"[HOM][4CRSRD][RVSON][2CRSRR]
OTRA[SPC]PARTIDA?[SPC](S/N)"
301 GETA$:IFA$="S"THENRUN
302 IFA$<>"N"THEN301
310 PRINT"[CLR]"CHR$(9):POKE36869,240:END
320 POKE36869,242:POKE36879,25:PRINT
CHR$(8)"[CLR][RVSON][BLK][4SPC][SHIFTL]
[SHIFTO][SHIFTS][SPC][SHIFTP][SHIFTI]
[SHIFTS][SHIFTT][SHIFTO][SHIFTL][SHIFTE]
[SHIFTR][SHIFTO][SHIFTS][4SPC]"
330 PRINT"[SPC][SHIFTJ]UEGO[SPC]PARA
[SPC]DOS[SPC]JUG.":PRINT"[CRSRD][RVSON]
[SPC][SHIFTM][SHIFTI][SHIFTS][SHIFTI]
[SHIFTO][SHIFTN][SPC][RVSOFF]:[SPC][SHIFTC]
ARGATE[SPC]AL"SPC(12)"CONTRARIO[SPC]
SI[SPC]"
340 PRINTSPC(10)"PUEDES."
350 PRINT"[2CRSRD][SHIFTF]UNCION[2SPC]
[SHIFTJ]UG.[SPC]1[SPC][SHIFTJ]UG[SPC]2"
360 PRINT"[CRSRD][SPC][SHIFTA]RRIBA[2SPC]
[RVSON][SPC]CTRL[SPC][RVSOFF][2SPC][RVSON]
[2SPC]↑[2SPC][RVSOFF]"
365 PRINT"[SPC][SHIFTA]BAJO[3SPC][RVSON]
COMM.[SPC][RVSOFF][2SPC][RVSON]CRSR←
[RVSOFF]"
370 PRINT"[SPC][SHIFTD]ISPARO[SPC][RVSON]
SHIFT[SPC][RVSOFF][2SPC][RVSON]CRSR↑
[RVSOFF]"
375 PRINT"[CRSRD][SHIFTP]ULSA[SPC][RVSON]
[SHIFTS][SHIFTH][SHIFTI][SHIFTF][SHIFTT]
.[SHIFTC][SHIFTT][SHIFTR][SHIFTL].[SHIFTC]
[SHIFTO][2SHIFTM]."
380 IFPEEK(653)<>7THEN380
385 RETURN
```

MASTER MIND

Javier Salas Varela
Velázquez, nº 1
Cádiz-10

C-64

Me llamo Javier Salas Varela y tengo 13 años de edad. Os mando este programa, pues yo también quiero aportar mi granito de arena a la revista. Espero publiquéis este juego "**MASTER MIND**". El programa esta realizado en un C-64.

Me lo sugirió mi hermana Milagros pues decía que como este popular juego no lo teníamos y yo me lancé y aquí está.

El programa no contiene muchos POKES por le que es muy fácil pasarlo al VIC-20. Explicación:

El ordenador te da a elegir si la combinación de números a adivinar sea de 1 a 6 dígitos, seguidamente nos pregunta si los dígitos se pueden repetir, se esperan unos segundos y saldrá la primera elección, el máximo de intentos es 100, quedando en el estabilizado hasta que se acierte, después el ordenador nos dice los puntos obtenidos en los cuales se tiene en cuenta el tiempo, el nivel, y otros factores, el ordenador si haces nuevo récord te lo dice y si no, te ofrece el anterior.

En fin buena suerte y que sigáis con esta magnífica revista. ■

```
10 REM**MASTER MIND**
15 DIM D(14)
20 POKE53281,1
30 PRINT"[CLR]"
35 TI$="000000"
40 PRINTTAB(14)"[RED][RVSON]MASTER[SPC]
MIND[RVSOFF]"
60 PRINT"[10CRSRD]":PRINT"[BLU][SPC]
REALIZADO[SPC]POR[SPC]J.[SPC]SALAS,P
ULSA[SPC]UNA[SPC]TECLA"
70 GET A$:IF A$=""GOTO 70
80 PRINT"[CLR][CRSRD][2SPC]EL[SPC]OR
DENADOR[SPC]ESCOGE[SPC]UN[SPC]NUMERO
[SPC]AL[SPC]AZAR."
90 M=0:N=0:B=0:RESTORE
100 PRINT"[2CRSRD]DE[SPC]CUANTOS[SPC]
TERMINOS[SPC]DESEA[SPC]EL[SPC]NUMERO
(1-6)"
105 INPUT A
107 IF A<1 OR A>6 THEN 80
109 D$="[4SPC]":FOR I=1 TO A:D$=D$+"
[BLK][SPC]":NEXT:FOR I=1 TO 40-2*A:D
$=D$+"[SPC]":NEXT
110 PRINT"[2CRSRD]DESEA[SPC]QUE[SPC]
SE[SPC]PUEDAN[SPC]REPETIR[SPC]TERMIN
OS"
120 INPUT"(S/N)";B$
125 IF B$<>"S" AND B$<>"N" THEN 120
130 PRINT:PRINT:PRINT"PULSE[SPC]CUAL
QUIER[SPC]TECLA"
135 GET C$:IF C$="" GOTO 135
137 IF B$="S" THEN GOSUB1000:GOTO140

138 GOSUB2000
140 GOSUB 800:PRINT"[CLR]":GOSUB5000

150 M=M+1:GOSUB 7000
160 FOR I=1 TO A
170 GET C$:IF C$="" THEN 170
180 IF C$<"1" OR C$>"9" THEN 170
190 C(I)=VAL(C$) :PRINTC$"[SPC]";
200 NEXT
205 GOSUB 6000
210 GOSUB 3000:IF N=A THEN 300
220 PRINTN"M"B"H":PRINT
225 IF M>8 THEN PRINT"[HOM]":GOSUB50
00:PRINT"[16CRSRD]";
230 GOTO150
300 PRINT"[CLR][2CRSRD][3CRSRR][RED]
";:FOR I=1 TO A:PRINT LEFT$(STR$(B(I
)),2);:NEXT
340 PRINT:PRINT:PRINT"HECHO[SPC]EN[SPC]
";M;"INTENTOS":GOTO10000
345 GOSUB 8000
360 GET H$:IF H$=""GOTO 360
365 IF H$="S" GOTO 70
370 END
510 INPUT C$(I)
520 NEXT I
530 RETURN
800 POKE 54296,9:FOR Z=1 TO 7:READ L
(Z):NEXT
810 FORX=1TO14:READ D(X):NEXT:FORJ=1
TO7:POKE 54275,L(J):FOR Z=1 TO D(2*J
-1):NEXT
```

```
820 POKE54275,9:FOR Z=1 TO D(2*J):NE
XT Z
830 NEXT J
840 DATA 195,195,195,195,195,195,201
850 DATA 80,60,60,60,60,60,60,100,80
,60,60,80,80,80
860 RETURN
1000 FOR I=1 TO A:B(I)=INT(RND(0)*9)
+1:NEXT I:RETURN
2000 FOR I=1 TO A
2010 B(I)=INT(RND(0)*9)+1
2020 FOR Z=0 TO I-1:IF B(I)=B(Z) THE
N 2010
2030 NEXT Z:NEXT I
2040 RETURN
3000 B=0:N=0:FOR I=1 TO A:IF C(I)=B(
I) THEN N=N+1
3005 NEXT
3010 IF N=A THEN RETURN
3020 FOR I=1 TO A
3030 FOR T=1 TO A:IF C(I)=B(T) THEN
B=B+1
3040 NEXTT:NEXT I:B=B-N:RETURN
5000 PRINT"[HOM][BLK]M=MUERTO[6SPC]H
=HERIDO[12SPC][RED]"D$"[BLU]":RETURN

6000 FOR I=1 TO 2:POKE 54296,3:POKE
54273,215
6010 FOR Z=1 TO 150:NEXT Z
6020 POKE 54273,0
6030 FOR Z=1 TO 150:NEXT Z
6040 NEXT I
6050 RETURN
7000 M$=STR$(N):PRINT MID$(M$,2,2)"
"TAB(4);:RETURN
8000 POKE 54296,3:FOR I=1 TO 3:POK
E54276,235
8010 FOR Z=1 TO 100:NEXT Z
8020 POKE54274,0
8030 FOR Z=1 TO 100:NEXT Z
8040 NEXT I
8050 POKE54295,220:FOR Z=15 TO 0 STE
P -1
8060 POKE54296,Z:FOR M=1 TO 100:NEX
T M
8070 NEXT Z
8080 POKE 54295,0:POKE54296,0
8090 RETURN
9000 FORI=1TOA:PRINTB(I),C(I):NEXT
10000 PRINT"HAS[SPC]TARDADO[2CRSRR]"
MID$(T$,3,2)"[2CRSRR]MINUTOS"
11000 I$=(MID$(TI$,5,2))
12000 PRINT I$"[2CRSRR]SEGUNDOS"
30000 PRINT"[CRSRD][13CRSRR][RVSON]H
A[SPC]CONSEGUIDO[RVSOFF]"
63901 PRINT"[CRSRD]" (((A+5046)+(
A*100)-(M*100)+(I+(Z)))"[CRSRR][RVSON]
PUNTOS[RVSOFF]"
63902 RR=(((A+5046)+(A*100)-(M*100
)+(I+(Z))))
63905 IF RR>SC THEN PRINT"[RVSON]NUE
VO[SPC]RECORD[RVSOFF]":PRINT"[13CRSRR]
[CRSRD]"RR:SC=RR
63945 PRINT"[RVSON]RECORD[RVSOFF]"SC

63999 PRINT:PRINT"QUIERE[SPC]JUGAR[SPC]
OTRA[SPC]VEZ?[SPC](S/N)[BLU]":GOTO36
0
```

EDITOR DE CARACTERES

Santiago Casas Duarte, 16 años
Emperatriz Eugenia 14, 1º
Terrassa BARCELONA. Tfno.: 788 04 33

VIC-20
SIN AMPLIACION

El programa que os presento es lo llamado UN EDITOR DE CARACTERES, ya que hasta el momento no he visto ninguno tan completo ¡¡modesto eh!! en la revista.

El primer consejo que os he de dar, es que una vez entrado el programa, lo grabéis, pues es un programa que se modifica él solo al ejecutarlo.

Al poner en marcha el programa, nos sale un interrogante en la esquina superior izquierda, esto significa que está esperando que entremos el carácter que debe modificar.

Una vez hecho se formará en la pantalla un cuadrilátero de 8×8 asteriscos excepto uno que es el signo "-", al que podemos llamar cursor.

Cuando apretamos la tecla f1 y movemos el cursor de sitio nos damos cuenta que el asterisco ha cambiado de color. Se está empezando a crear un carácter. La tecla f7 borra el punto.

Cuando apretamos la tecla f5 nos sale el dibujo que estamos formando mucho más claro. Cuando la tecla que apretamos es la f6, y esperamos un poco (faltaría!) sale el carácter que queremos editar a tamaño natural. Apretamos una tecla y volvemos al estado normal.



Una vez creemos que el carácter ya está bien diseñado, apretamos la tecla f3, lo cual nos crea una instrucción DATA, volviendo a preguntar por otro carácter.

Una vez hayamos terminado de editar caracteres, apretamos la tecla f4 y desaparecerá el programa del editor quedando sólo las instrucciones DATA y un pequeño programa, que poniéndolo delante de un programa en el cual necesitáis los caracteres creados y escribiendo POKE 36869,255 los podréis utilizar, y para volver al modo normal del VIC, escribir POKE 36869, 240.

Otra cosa, el programa está preparado para funcionar sin ningún tipo de ampliación de memoria. ■

```
10 NL= 9000
20 X=1:Y=1
30 DIML(8,8),A(8)
40 AB$="[8CRSRD]":AL$="[8CRSRR]"
50 POKE36879,8
60 PRINT"[CLR]?":GETA$:IFA$=""THEN60
70 PRINT"[HOM]"A$:WB=ASC(A$):IFWB=>6
4THENWB=WB-64
80 FORI=1TO8:PRINT"[SPC]********"
90 NEXTI
100 PRINT"[HOM]";LEFT$(AB$,Y);LEFT$(
AL$,X);"-"
110 GETA$:IFA$=""THEN110
120 IFA$="[F7]"THENL(X,Y)=0:GOTO140
130 IFL(X,Y)=1THENPRINT"[HOM]";LEFT$
(AB$,Y);LEFT$(AL$,X);"[WHT]*[BLU]":G
OTO160
140 IFA$="[F1]"THENL(X,Y)=1:GOTO160
150 PRINT"[HOM]";LEFT$(AB$,Y);LEFT$(
AL$,X);"*"
160 IFA$="[CRSRR]"THENX=X+1
170 IFA$="[CRSRL]"THENX=X-1
180 IFA$="[CRSRD]"THENY=Y+1
190 IFA$="[CRSRU]"THENY=Y-1
200 IFA$="[F4]"THENGOTO720
210 IFA$="[F3]"THEN500
220 IFA$="[F5]"THENGOSUB290
230 IFA$="[F6]"THENGOSUB370
240 IFX>8THENX=1
250 IFX<1THENX=8
260 IFY>8THENY=1
270 IFY<1THENY=8
280 GOTO100
290 FORJ=1TO8
300 FORI=7690+22*JTO7697+22*J
310 H=H+1
320 IFL(H,J)=1THENK=160:GOTO340
330 K=32
340 POKEI,K:POKEI+30720,1
350 NEXTI:H=0:NEXTJ
360 RETURN
370 FORI=1TO8:A(I)=0:NEXTI
380 TG=1:FORJ=1TO8
390 FORI=8TO1STEP-1
400 IFL(I,J)=1THENA(J)=A(J)+TG
410 TG=TG*2
420 NEXTI:TG=1:NEXTJ
430 POKE52,28:POKE56,28
440 FORI=0TO511:POKE7168+I,PEEK(3276
8+I):NEXTI:POKE36869,255
450 FORI=7416TO7423:GH=I-7415:POKEI,
A(GH):NEXTI
460 PRINT"[HOM][14CRSRD][5CRSRR][WHT]
←"
470 GETB$:IFB$=""THEN470
480 PRINT"[HOM][14CRSRD][5CRSRR][3SPC]
":POKE36869,240
490 RETURN
500 REM CONTRUIR DATA
510 FORI=1TO8:A(I)=0:NEXTI
520 TG=1:FORJ=1TO8
530 FORI=8TO1STEP-1
540 IFL(I,J)=1THENA(J)=A(J)+TG
550 TG=TG*2
560 NEXTI:TG=1:NEXTJ
570 PRINT"[CLR][2CRSRD]";
580 PRINTNL;"DATA";WB",";
590 FORI=1TO8
600 PRINTA(I);
610 IFI<8THENPRINT"[CRSRL],";
620 NEXTI
630 PRINT
640 PRINT"10[SPC]NL=";NL+1
650 PRINT"RUN[BLU]"
660 PRINT"[HOM]";
670 POKE198,3:POKE649,10
680 FORI=631TO633
690 POKEI,13
700 NEXTI
710 END
720 PRINT"[CLR][CRSRD]"
730 PRINT10000"G1=32768:G2=7168:FORI
=0TO511"
740 PRINT10001"POKEG2+I,PEEK(G1+I)"
750 PRINT10002"NEXTI"
760 PRINT10003"FORI=1TO"NL-9000":REA
DC:FORJ=0TO7:READK"
770 PRINT10004"POKEG2+(C*8)+J,K:NEXT
J,I:POKE52,28:POKE56,28
780 PRINT"RUN850"
790 PRINT"[HOM]";:POKE198,6:FORI=631
TO636:POKEI,13:NEXTI:END
840 PRINT"[HOM]";:POKE198,9:FORN=631
TO640:POKEN,13:NEXT:END
850 N=0
860 PRINT"[CLR][CRSRD]":FORM=NTON+60
STEP10:PRINTM:NEXT:PRINT850"N="M:PRI
NT"RUN850"
870 IFN<785THEN840
880 PRINT"[CLR][2CRSRD]"
890 FORN=840TO890STEP10:PRINTN:NEXT:
PRINT"POKE1,0:SYS1":GOTO840
```

El Misterio de los Pokes (V)

Por Diego ROMERO

Como en números anteriores, vamos a seguir avanzando dentro del mapa de memoria de nuestros microordenadores. Primero veremos la tabla y luego la función de cada una de las posiciones de memoria.

Dirección		Nombre	Función
Hex	Dec		
39-3A	57-58	CURLIN	Número de línea actual
3B-3C	59-60	OLDLIN	Número de línea anterior.
3D-3E	61-62	OLDTXT	Puntero para CONT
3F-40	63-64	DATLIN	Número de línea DATA actual.
41-42	65-66	DATPTR	Puntero término DATA actual.
43-44	67-68	INPPTR	Puntero de INPUT.
45-46	69-70	VARNAM	Nombre de variable.
47-48	71-72	VARPNT	Puntero a dato de variable
49-4A	73-74	FORPNT	Puntero de lazos FOR-NEXT.
4B-60	75-96	TEMPTR	Punteros temporales.
61	97	FACEXP	Exponente del FAC número 1.
62-65	98-101	FACHO	Mantisa del FAC 1.
66	102	FACSGN	Signo acumulador de coma flotante.
67	103	SGNFLG	Bandera de evaluación de signo.
68	104	BITS	Bandera de sobrecarga de FAC.
69	105	ARGEXP	Exponente del FAC número 2.
6A-6D	106-109	ARGHO	Mantisa del FAC 2.
6E	110	ARGSGN	Signo del FAC 2.
6F	111	ARISGN	Signo de operación entre FAC 1 y 2.
70	112	FACOY	Sobrecarga y redondeo de FAC.



Los punteros CURLIN y OLDLIN, son como podéis ver, los que contienen el número de línea que se está ejecutando en ese mismo momento y el de la última línea ejecutada respectivamente. Pueden ser útiles para que interceptando la ejecución de las rutinas de interrupción por medio de un programita en código máquina, podamos visualizar en una esquina de la pantalla qué línea se está ejecutando, o en caso de error cuál fue la línea anterior al error, etc.

El puntero OLDTXT, señala el último comando o instrucción BASIC que se estaba ejecutando, sirve para poder utilizar el comando CONT y continuar la ejecución de un programa después de una parada de programa (con STOP).

DATLIN contiene el número de la línea data de la que se están leyendo los datos con READ; se puede utilizar para saber en qué línea nos hemos equivocado al copiar un programa, podéis ver cómo hacerlo en el programa "teclas de función" (número 3 de la revista).

DATPTR es el puntero que señala la posición de memoria donde se encuentra el dato que vamos a leer con una sentencia READ. Podemos alterarlo junto con el anterior para simular una sentencia RESTORE NNNN, ya que la que tenemos en nuestros equipos lo hace al primer DATA, pero no podemos hacerlo a un número de línea determinado.

INPPTR, es un puntero que utiliza la rutina de entrada de datos 'INPUT', para leer los datos que nosotros hemos tecleado detrás de la interrogación que aparece en la pantalla.

VARNAM es el nombre de la última variable a la que nos hemos referido en el programa BASIC.

VARPNT es el puntero al dato que se está utilizando actualmente en la variable BASIC.

FORPNT es el puntero que señala a la variable utilizada como índice del último lazo FOR-NEXT abierto.

TEMPTR son un conjunto de punteros utilizados para diversas aplicaciones por el intérprete BASIC, no tienen una aplicación fija en exclusiva. Podéis ver como ejemplo la rutina "Move Memory" aparecida

en la sección de magia del número 6 (Página 51).

FACEXP es el exponente del acumulador de coma flotante número uno. FACHO es la mantisa del mismo acumulador, y FACSGN es su signo.

SGNFLG es un byte que sirve para calcular el signo de una operación, mientras que BITS, actúa como indicador de sobrecarga de la operación en el acumulador, si los valores alcanzados se han salido de los límites permitidos.

ARGEXP es el exponente del acumulador número dos, ARGHO es su mantisa y ARGSGN es su signo.

La función de ARISGN es la de calcular el signo del resultado de la operación entre los dos acumuladores, y la de FACOV, comprobar si hubo sobrecarga o redondeo en el resultado.

Los acumuladores de coma flotante

Para almacenar en la memoria los números de nuestro sistema de numeración (decimal), y poder realizar con ellos operaciones matemáticas, los microordenadores se ven obligados a convertirlos en sus equivalentes binarios en notación de coma flotante.

El lugar donde el ordenador los maneja, los convierte de decimal a binario en coma flotante y viceversa, y realiza las operaciones aritméticas y lógicas, se denomina acumulador, y para diferenciarlo del que utiliza la CPU, especificamos que es de coma flotante, respondiendo a las siglas de FAC (Floating Point ACcumulator).

En los acumuladores de coma flotante o FAC, la información se guarda según el formato de la figura 1.

Podemos ver que en la primera posición de memoria aparece el exponente, este corresponde al utilizado en la notación científica, que es en realidad la utilizada para los FAC. También vemos que al exponente se le suma 129 para que se puedan almacenar exponentes positivos y negativos.

La mantisa del número almacenado en el FAC 1 se guarda en las posiciones \$62 a \$65, denominadas FACHO, y su signo se calcula en la siguiente posición (FACSGN), pero se guarda como el bit más alto (el séptimo) del primer byte. La siguiente dirección de memoria se comporta como byte de evaluación de signo para las operaciones que se efectúen con el FAC. Finalmente BITS, sirve como indicador de sobrecarga, tanto por exceso como por defecto, en todos los cálculos que realiza el intérprete BASIC con el acumulador número 1. El error "overflow", aparecería si el número almacenado en el FAC fuese mayor o menor que las cantidades máxima o mínima que puede manejar el FAC.

Las siguientes direcciones de memoria, se comportan exactamente igual que estas últimas pero para el segundo acumulador FAC, que es el que se utiliza como argumento en la mayoría de las operaciones. ■

Exponente binario +129	Mantisa binaria en BCD empaquetado dando una precisión de 8 dígitos. El primer bit del primer byte es el signo

Magia



La MAGIA son trucos, la MAGIA es divertida.

La MAGIA es hacer lo que nadie se ha atrevido y resulta ser la fuente más completa de información para la informática práctica.

La MAGIA es una sección mensual llena de consejos, trucos, de esto y aquello del mundo del software, hardware y aplicaciones, trucos descubiertos por los demás que hacen que la informática sea más fácil, más divertida o más animada.

MAGIA habla de ideas sencillas, programas de una sola línea, subrutinas útiles, hechos de informática poco conocidos y otras cosas de interés.

PREMIOS: *(VER PAG. 4 y 5)*

Recuperador

Cargando un programa desde el cassette se puede dar el caso en que el ordenador nos dé el mensaje de ? LOAD ERROR.

Comprobamos, sin embargo, que el listado aparece íntegro pero si hiciéramos RUN nos quedaríamos sin el programa.

Si el error de carga se sitúa hacia el final de la grabación del programa es posible (pero no siempre) recuperarlo tecleando la siguiente línea en modo directo:

POKE45 , PEEK(174) : POKE46 , PEEK(175) : CLR (RETURN)

Es condición indispensable el no haber hecho RUN después de la operación de carga.

Si el error se da en la carga de un fichero de datos, que puede formar parte de un programa este método no es aplicable.

La mejor recomendación es sin duda el tener una copia de seguridad de cada programa y recurrir a ella en estos casos.

Si paramos la carga de un programa con la tecla RUN/STOP (en la segunda mitad de la grabación del mismo) y después tecleamos la línea antes citada podemos comprobar que al hacer RUN el programa se ejecuta correctamente. Se puede dar algún caso en que esto no sea cierto ya que estamos pasando por alto la seguridad de la carga de programas desde el cassette.

J. R. L. (Iniciales por expreso deseo)
BERGARA (Guipúzcoa)

Acelera la programación

Tengo un VIC-20 y soy suscriptor de vuestra revista. Se nota un importante avance en su contenido a medida que van apareciendo nuevos números. Os felicito por ello.

En cuanto a la sección de Magia, también quiero hacer mi pequeña aportación:

1) Conecte el VIC-20 y prepare la siguiente línea:
PRINT 2+2 y pulsando RETURN aparece 4.

También preparando la línea: ?2+2 y pulsando RETURN aparece 4.

2) Ahora, prepara la siguiente línea: 10 PRINT "Y" y pulsando RETURN y luego tecleando RUN sele Y. También, preparando línea: 10?"Y" y pulsando RETURN y luego tecleando RUN sale Y, y además tecleando LIST sale 10 PRINT "Y".

Conclusión: El símbolo ? equivale al PRINT. Por tanto se consigue un ahorro de teclado (4 símbolos).

Esto es todo por ahora. Espero continuar participando.

Joan Mons i Bruguera. 10 años
C/María Vidal, 20. Vilassar de Mar. Barcelona.

Tabla de multiplicar
Bandera de España

Esta es la segunda vez que os escribo, y en esta ocasión aprovecho para enviaros un programa de una sola línea para la sección de Magia y con el cual se puede conseguir la tabla de multiplicar del 1 al 10 de cualquier número, es muy sencillo, aunque espero que sea de vuestro interés para publicarlo.

También añado otro programa, éste es de Fco. Federico Iborra de Madrid que salió publicado en la revista 4 del mes de mayo y que dibujaba la bandera española, el cual tuve que rectificar, pues intenté introducirlo en mi VIC-20 con los cambios de POKE que apuntaba y el color amarillo no aparecía. Con el que envío, sí que se dibujó la bandera con sus dos colores —rojo y amarillo—, pero para que estuviese dentro de una línea de programa tuve que abreviar todas las sentencias, pues de otra manera no resultaba.

Las variaciones que efectué fueron las siguientes:

Cambié: POKE 34304+A,7 por 38532+C,7, para que apareciese el color amarillo.

Añadí: FORC=Oto241 para que el color amarillo ocupara la franja central de la bandera.

Y pasando a otra cosa y si no es mucho preguntar, quisiera saber la longitud de bytes que tienen los dos programas y cómo se averigua.

Tabla de multiplicar:

```
1 PRINT"TABLA[SPC]DE[SPC]MULTIPLICAR ":INPUTA:FORC=1
TO10:B=A*C:PRINTA; "X";C;"=";B:NEXT:GOTO 1
```

Bandera de España:

```
1 PRINT"[CLR]":POKE36879,41:FORC=0TO 241:POKE38532+C,7:
NEXT:FORA=7812TO80 53: POKEA,102;NEXT:FORB=1TO2199:NEXT
```

Lourdes Esteban Díez
C/Lope de Vega, 265, At. 1º Barcelona-18

Utiliza REM

Si se quiere eliminar una línea, o parte de ella, de un programa para ver su efecto sobre el mismo, pero luego se quiere recuperar sin escribirla de nuevo, se escribe un REM delante de la porción de línea, o línea que queremos borrar; al encontrarse el ordenador con un REM, salta a la línea siguiente y no ejecuta lo que hay detrás del REM.

Otro: Si estamos usando el juego de caracteres del usuario y por ejemplo tenemos un gráfico definido en la letra A y necesitamos usar esta letra y el gráfico a la vez, sencillamente cuando queramos usar la A, la ponemos en RVS/ON, y para el gráfico RVS/OFF.

Javier Sebastián Segura, 17 años.
C/Puebla de Valverde 8-25. Valencia-14

Rutina Merge

Para cambiar el inicio de Basic y poder mezclar programas de disco o de cinta en el Commodore 64, hay que tener en cuenta que las líneas del programa que se añade tienen que ser más altas que el anterior programa y hacer poke 43,1: poke 44,8 run antes de unir el siguiente programa.

Ejemplo:

Load "Merge" 8,1 (Nombre de la rutina)

Run (nos imprimirá POKE43,???:POKE44,???), entonces si posicionamos el cursor en la línea de texto y presionamos return se cambia el inicio del basic lo podemos comprobar con list y no aparecerá el programa si esto es así cargamos la primera parte del programa con Load "programa", 8, luego pondremos POKE43,1:POKE44,8, después run, saldrá de nuevo

```
POKE43,???:POKE44,???,
```
y lo mismo que antes con list no aparece ningún listado luego otra vez load "programa", 8 etc... Cuando se termine de añadir programas hacer POKE43,1:POKE",8:LIST y borrar las líneas 1-6 que son del programa Merge, después guardarlo si se quiere.

```
1  I1=1: I2=8: L1=I1: L2=I2: GOSUB5
2  L1=PEEK(AD):L2=PEEK(AD+1):GOSUB5
3  IFL1=0ANDL 2=0THENL1=X1:L2=X2:GOSUB5:GOTO 6
4  X1=L1:X2=L2;GOTO2
5  AD=0:AD=L2*256+L1:RETURN
6  PRINT"[CLR]POKE43,"L1;":POKE44,"L2;END
READY
```

José Manuel Moreno Tobaruela
C/Navarro Villoslada, 10-3º Dcha. Pamplona

Goma de borrar

Os envío un truco (si es que se puede llamar así), que sirve para borrar de la memoria un programa cuando te dan a elegir entre continuar en él o no.

La instrucción es: SYS 58232

Con ella, en la pantalla del VIC aparece lo mismo que si lo hubieras apagado y encendido.

Luis Alfonso Sanz
Pº San Isidro, 5. 2ºA. Valladolid-12

Nota: La instrucción correcta para simular totalmente el apagado es: SYS 64802 para el VIC-20 y SYS 64738 para el C-64

Instrucción Auto C-64

Os envío un programa para la sección de Magia que puede interesar a los usuarios del Commodore 64. Es una rutina en lenguaje máquina que realiza la instrucción "AUTO", la cual los ordenadores Commodore no poseen. Esta instrucción genera automáticamente los números de las instrucciones correlativas de un programa en Basic. O sea, el programa empieza en la línea 100, y el número se va incrementando automáticamente de 10 en 10 (100, 120, 130, etc). Para salir de la rutina sólo se puede hacer pulsando RETURN en una instrucción en blanco, y para volver a ella basta teclear:

SYS 49152

Pero de esta manera volverá a empezar a la instrucción 100, por lo que debéis pulsar RUN/STOP y RESTORE simultáneamente, repetidas veces, y aparecerán los números de instrucción en la parte superior izquierda de la pantalla, hasta que lleguéis a la instrucción en la que queréis continuar. Aunque aparezcan los números solos, esas instrucciones no se borran. La rutina empieza en la localidad 49152 de memoria. Para utilizar el programa después de haberlo tecleado, se debe almacenar en cassette o en disco, porque al ejecutarse el programa se borra automáticamente (línea 50). Antes de empezar a programar se debe cargar y ejecutarlo y el número de instrucción 100 aparecerá en la pantalla. Si se desea que el número de incremento sea diferente de 10, basta POKEar en la localidad de memoria 49179 el número de incremento deseado.

```
10 X=49152
20 READ Y:IF Y=-1 THEN 40
30 POKE X, Y: X=X+1: Z=Z+Y: GOTO 20
40 IF Z    12374 THEN PRINT "ERROR EN LAS SENTENCIAS
DATA":END; REM si Z es diferente de...
50 SYS 49152: NEW
100 DATA 169,90,133,251,169,0,133,252,169,19,14,2,3,169,192,141,3,3,
    96,32,25.
110 DATA 192,76,134,164,24,169,10,101,251,133,251,144,2,230,252,165,
    251,133,99
120 DATA 165,252,133,98,162,144,56,32,73,188,32,221,189,162,0,189,1,1,
    240,9,32
130 DATA 210,255,157,0,2,232,208,242,32,18,225,201,13,240,3,76,105,165,
    56,165
140 DATA 251,233,20,176,2,198,252,169,131,141,2,3,169,164,141,3,3,76,
    118,165,1
```

Podéis resumir las instrucciones de manejo donde os parezca conveniente.

Pedro Obrador Espinosa
C/Costa I Llobera, 12. Arta (Mallorca)

Quinielas de 14

Primero os quiero felicitar por la revista tan buena que hacéis. Me llamo Javier Recio Lamata y tengo un VIC-20, soy lector de la revista y vi que un lector quería un programa para hacer quinielas, pues bien yo lo tengo y os lo envío:

```
10 FOR A=1 TO 14
20 L=INT (RND(1)*10)+1
30 IF L<=5 THEN PRINT "PARTIDO";A;"-SIGNO 1"
40 IF L>=5 AND L<9 THEN PRINT "PARTIDO";A"-SIGNO X"
50 IF L>8 AND L<=10 THEN PRINT "PARTIDO";A;"-SIGNO 2"
60 NEXT A
70 END
```

Este programa da una probabilidad del 50% para el signo 1, del 30% para el signo X y el 20% para el signo 2, se simula el lanzamiento de un dado de 10 caras.

Javier Recio
Fernán Caballero, nº 22. Sevilla-1.

Pajaritos

Estas líneas se pueden utilizar como subrutina cuando se necesite un sonido de pajarillos en el VIC-20 con el SUPEREXPANDER.

```
1Ø FOR K=1 TO 2Ø
2Ø FOR L=254 TO 24Ø+INT(RND(1)*1Ø) STEP -1
3Ø SOUND Ø,Ø,L,Ø,15
4Ø NEXT L
5Ø SOUND Ø,Ø,Ø,Ø,Ø
6Ø FOR L=Ø TO INT(RND(1)*1ØØ)+12Ø
7Ø NEXT L
8Ø NEXT K
```

Francisco Saez
C/Virgen del Rosario, 28. Alcantarilla (Murcia)

VIC-Travesti

Envío un pequeño programa para el VIC-20 que se podía titular **"VIC Travesti"**, pues al hacer RUN no sólo cambia al imprimirlo el nombre de hombre del programa a otro de mujer sino que lo deja modificado en el mismo.

```
10 A$="JUAN":PRINT(CLR):A=256*PEEK(642)
20 FOR N=1 TO 4: READ B: POKEA+8+N,B:NEXT;PRINT A$
30 DATA 76, 79, 76, 65
```

Federico Iborra
C/Comandante Zorita, 8. Madrid-20

De un Commodoriano

Os envío un truco para nuestra querida sección de magia. Ahi va:

Si os molesta el parpadeo del cursor en el VIC, y os gustaría hacerlo desaparecer, simplemente teclead:

POKE 788,225. El cursor ya no parpadeará.

Si por el contrario queréis que desaparezca, escribid:

POKE 788,255

Si queréis borrar la pantalla (como haríais con un run stop/restore) Teclead:

POKE 789,0

Que, ¿os han gustado las magias?

P.D. No os quejéis, si me he "pasao" haciendo preguntas, pero es que es la primera vez que escribo.

"adeu"

Leonardo García Font
C/Centro nº 21. Piso 1º 1ª. Barcelona-26

Pokes de Control

Queridos VIC-ciosos, mando a continuación las direcciones de memoria para el VIC-20, con distintas ampliaciones, que la hacen variar:

3k-1024;4095

más de 8 u 8k -4096;7679 (la memoria de color pasa a -37888;38399).

Espero que si compráis una ampliación de memoria no se os ocurra como a mí que no podía utilizarla en muchas ocasiones.

Mando también unos POKES muy útiles:

POKE 650,128: repetición de todas las teclas
POKE 650,100: anula dicha repetición
POKE 646 nº del color: color del cursor (0=negro, 1=blanco...)
POKE 656,128: simula RUN/STOP-RESTORE

Gonzalo Vidal
C/Tilos, 29. Madrid-23

"Función seno"

El presente mini-programa dibuja en el VIC-20 la función seno; si bien la resolución no es muy buena, que digamos, ahí va, por si a alguien le interesa:

```
1Ø PRINT "[CLR]": POKE 36879, 11Ø: FORN=ØTO21:
POKE 7922+N+22*INT (11*SIN(N*π/1Ø,5)), 42:
NEXT
```

Si queréis aumentar la frecuencia no tenéis más que multiplicar por 2 el numerador del argumento de la función SENO. Y si queréis aumentar la resolución, colocar un STEP.5 en el bucle FOR...NEXT.

Espero que os guste.

Octavio Navarro Caceres, 17 años.
Ctra. Fuente Encina, 101, 2º 3º.
Sta. Coloma de Gramanet (Barcelona). Tfno.: 391 69 50

Decimal a Binario

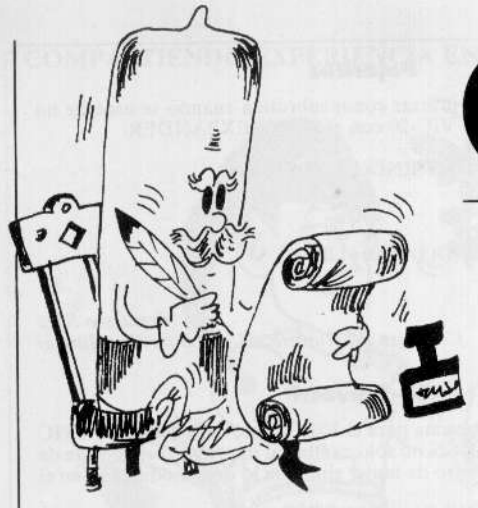
Aquí os va un programa de una sola línea que transforma un número decimal en binario.

```
1Ø INPUT N: FOR A=Ø TO 7: S=2↑A: W$=RIGHT$ (STR$((N AND S)
/S,1)+W$: NEXT: PRINT W$
```

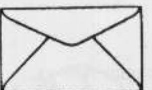
Como véis es muy sencillo, a ver si lo podéis hacer más corto.

Miguel Angel Folgado Costa
Avda. Blasco Ibáñez, 29, 1º. Manises (Valencia).

CARTA BLAN



UTILIZAR EL JOYSTICK



Hola, hace poco os mandé un truco de magia y ahora os ruego que me hagáis un favor. Hace unos días compré un Joystick y lo probé con dos o tres programas que saqué de unas revistas. Al cabo de unos días decidí adaptar unos juegos para moverlos con el Joystick. Miré las revistas en el número 6 y vi los PEEKs para leer el movimiento del Joystick, y con ayuda de los programas anteriores, intenté adaptar los nuevos programas o intentar hacer yo uno: no lo conseguí. Creo que sería adecuado dedicar un artículo en vuestra revista a la utilización del Joystick y la rutina necesaria para que el ordenador lo lea.

CHARLY - (BARCELONA)

Tomamos nota de tu petición, no obstante como ya habrás visto lo único que necesitas para leer el joystick es hacer PEEK(56320) o PEEK(56321), dependiendo del PORT que vayas a usar (el 1 ó el 2).

MANUAL DEL SUPEREXPANDER



Soy José Forns un suscriptor de vuestra revista, y ahora me dirijo a vosotros por tener un grave problema con el cartucho del superexpander. Este problema se refiere a la ROM de éste y es que hay una infinidad de comandos que no sé para qué sirven ni cómo se utilizan, estos son: SCNCLR, SOUND, RGE, RCOLOR, RDOT, RPOT, RPEN, RJOY y RSND.

De estos comandos, la verdad, sé que existen porque vosotros en vuestro anuncio de contraportada de algunos números los exponéis pero si no los hubierais puesto ni los conocería. Luego hay otros comandos que me suenan y sé para qué sirven pero no los sé usar pues al utilizarlos me dan error de sintaxis: estos son: KEY, COLOR, PAINT.

En este último en concreto no sé cómo pintar sectores circulares sin que invadan los sectores circulares colindantes pintados de otro color.

Me gustaría también que me contarais si es que es posible el grabar en cinta únicamente lo que es la pantalla en un programa, por ejemplo del programa de gráficas tridimensionales el poder grabar sólo el dibujo ahorraría muchíííísimo tiempo. Bien, no os entretengo más, esperando que me publiquéis la respuesta se despide vuestro amigo.

JOSE FORNS ANELO
C/VILAMARI, 33, 3º 1ª.
BARCELONA.

Como ya habrás visto, en el número 6, comenzamos a publicar el manual del superexpander en castellano. Espero que esto te solucione tus problemas.

Para grabar en cinta sólo el contenido de la pantalla, deberás hacerlo desde una rutina de código máquina similar a la publicada en el número 6 ("salvaguarda de memoria en periféricos"), pero con distintas posiciones de memoria, ya que esta se coloca a partir de la $C000, y tú deberás colocarla en donde tengas RAM.

PRECIO, PASCAL Y PET SPEED



Soy uno de los antiguos suscriptores de CLUB COMMODORE desde el nº 10 aunque los tengo todos, también soy uno de los que compraron un CBM-64 por 110.000 ptas. para observar con consternación que una semana después su precio pasaba a ser de 79.000.

Cuando compré el ordenador, en la propaganda aparecía: ...lenguajes que puede utilizar con UCSD PASCAL... También hay a su alcance un compilador de BASIC llamado PET SPEED.

En la propaganda que aparecía en Club Commodore decía: ...están en fase de desarrollo asimismo otros lenguajes como Ucsd Pascal. Y ya habían desaparecido las referencias a Pet Speed.

En la propaganda actual siguen estando en fase de desarrollo algunos lenguajes, pero el Pascal no ¿Esto qué significa, ya está desarrollado y lo veremos aparecer próximamente, o no sabremos nunca más de él?, ¿Salió de nuestro alcance el Pet Spedd? Agradecería me dieran respuesta a estas preguntas, o se pudieran poner en contacto con MEC para que ellos las respondieran, junto con estas otras.

¿Para cuándo la extensión del Basic que ofrecía cien nuevas sentencias al lenguaje normalizado, entre ellas sentencias estructuradas, generadores de sonido y gráficos simplificados?

Quizás haya salido y esté en el catálogo con otro nombre, de ser así, me gustaría conocerlo.

Ahora unas preguntas de carácter más técnico:

¿Qué significa la R de cp/m (r)? Es este CP/M el standard compatible con Apple, Sharp, Osborne, o cualquier otro aparato que posea CP/M 80?

¿Es imprescindible o muy necesaria la posesión de 2 drives para poder ejecutar con normalidad los programas escritos bajo este standard?

¿Es posible utilizar simultáneamente en CP/M mayúsculas y minúsculas? algunos lenguajes como el Prolog, así lo exigen.

¿Hay alguna posibilidad de ampliar el formato de pantalla a 80 columnas?

¿Sigue siendo posible acceder bajo el CP/M a la alta definición y los sprites?

Sé que la carta es larga y las preguntas muchas, pero considero que las respuestas, ofrecerán mucha luz a usuarios como yo que desean sacar el máximo rendimiento a su aparato, de acuerdo a las especificaciones que Commodore ofrece de su aparato.

Se despide:

LUIS PARIS GARCIA
CTRA. VIC 119-121
MANRESA (BARCELONA)

En el número anterior, aparecieron dos cartas contestadas por MEC que pueden contestar a tus preguntas, pero como nosotros hemos conseguido referencias de casas que disponen del lenguaje PASCAL para el C-64 fuera de España, te damos las señas que para puedas pedir información:

—La casa ABACUS Software de Grand Rapids (Michigam), tiene comercializado desde hace poco más de un año un compilador PASCAL para el C-64 denominado "ZOOM Pascal 64", según la información de que disponemos es rápido ya que produce código máquina directamente ejecutable por la CPU, no pseudocódigos. El paquete está compuesto por un editor (para crear y modificar programas en Pascal), un compilador que traduce el Pascal a un código intermedio, y un convertidor que pasa el código intermedio a código máquina. Además viene un manual completo y varios programas de demostración.

—En mayo del 84, la casa Data Becker ha lanzado una nueva versión, la 3,0, de su Pascal para el C-64. De momento sólo

hemos conseguido tener la noticia de su lanzamiento, pero intentaremos conseguir más información, si deseas pedirla tú mismo puedes hacerlo a la casa en Alemania:
DATA BECKER
Merowinger str. 30
4000 Dusseldorf
Tel.: 07-49-211-310010

El Pet Speed no ha "salido de nuestro alcance", se trata de un programa de Oxford Computer Systems ltd., Hensington Road, Woodstock Oxon. Tel.: 07-44-993-812700.

La extensión al Basic que prometió Micro Electrónica y Control está en el mercado desde hace varios meses (lo vi en distribuidores a primeros de mayo estaba). Su nombre es Simon's Basic.

La R en el nombre es para indicar que se trata de una marca registrada (en este caso por Digital Research). Es compatible en tanto en cuanto es CPM, pero esto no quiere decir que un disco del Apple lo puedas cargar en un Osborne ni en un C-64, ya que los formatos de grabación en disco son distintos.

Los dos drives no son imprescindibles para utilizar el CPM del C-64, pero es difícil trabajar CPM del C-64, pero es difípodrás comprenderlo si te compras cualquier libro de introducción al CPM y lo estudias un poco.

Las mayúsculas y minúsculas funcionan igual que normalmente. Lo de los sprites no te lo puedo contestar porque no lo he probado.

JUEGOS PARA EL 700



Amigos de Commodore World, antes de todo, quiero felicitaros por esa magnífica revista que habéis creado. Os envío tres faenas. Ahí van:

1. ¿Existe algún tipo de software (juegos, etc.), aparte del soft seriote para el CBM-700? ¿Puede crear gráficos?

2. ¿Puede ampliarse realmente el VIC-20 a 64k? En el número 5 de Commodore World, pusisteis un anuncio que hacía referencia a eso, pero he preguntado en algunas tiendas importantes y no saben nada del asunto. Si fuera así, se acercaría a la compatibilidad con el C-64? Otra cosa, ¿cuál es mi número de socio?

3. Y van tres. Como un Vic-cioso que soy, os envío un truco para nuestra querida sección de magia. Ahí va:

Si os molesta el parpadeo del cursor en el VIC, y os gustaría hacerlo desaparecer, simplemente teclead:
POKE 788,225. El cursor ya no parpadeará.

Si por el contrario queréis que desaparezca, escribid:
POKE 788,255

Si queréis borrar la pantalla (como haríais con un run stop/restore)

Teclead:
POKE 789,0

Qué ¿os han gustado las magias?

P.D. No os quejéis, si me he "pasado" haciendo preguntas, pero es que es la primera vez que escribo.
"adeu"

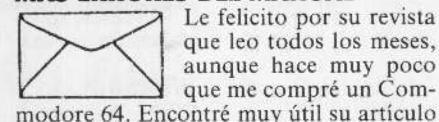
LEONARDO GARCIA FONT
C/CENTRO Nº 21. PISO 1º 1ª.
BARCELONA

1. No tenemos ninguna lista de juegos para el 700, no hay ningún motivo por el que no se puedan diseñar juegos, de hecho el controlador de vídeo que utiliza el CBM 700 puede dar un número de pixels mucho mayor que el del Commodore 64. La única causa que limita la utilización del 700 para juegos es su precio, si piensas comprarte un ordenador para jugar, ¿no te parece demasiado caro? Quien compre este ordenador seguro que no ha pensado en él para dedicarlo a juegos.

2. Estamos esperando que esta casa nos envíe la tarjeta para hacerles la reseña, cuando lo hagamos te podré contestar más seguro, de momento te puedo decir que el VIC-20 puede direccionar hasta 40 KRAM, no sé si al resto accede por medio de conmutaciones de bancos de memoria.

3. Hemos pasado tu truco a la sección de magia.

MAS ERRORES DEL MANUAL



Le felicito por su revista que leo todos los meses, aunque hace muy poco que me compré un Commodore 64. Encontré muy útil su artículo "Comando Old" del nº 3.

Leyendo el "Manual del Usuario" del Commodore 64, he encontrado algunos errores.

En la pág. 78, en la línea 30 falta un 2.

En la pág. 86, las explicaciones de las líneas del programa están equivocadas.

En la pág. 88, en la línea 20 debería poner READF.

La línea 460 de la pág. 145 es innecesaria, ya que es igual que la 500.

En la línea 630 de la pág. 146, el "3" debería de ser "S".

En la pág. 156, en el Registro nº 23, debe leerse AMPLIACION SPRITE Y.

En la pág. 147, la variable K$ de la línea 220 tiene mal sus 5 últimos términos. Estos tienen que ser "@-*£ ↑". En la línea 280 hay un extraño REM+16+64, que impide al programa ejecutar el POKE que le sigue. Por último no entiendo para qué sirve la subrutina que comienza en la línea 800. El programa nunca la ejecuta.

ALFREDO TORRES SAIZ
C/CENIA Nº 6 1º D
50016 - ZARAGOZA

Gracias por enviarnos todos los errores que has encontrado, de este modo podemos ir publicándolos y servirán de ayuda a otros usuarios de los equipos Commodore.

Imagino que la rutina de la línea 800, debería ser la que utilizó al principio el autor del programa en lugar de la línea 300, no sé por qué no la quitó del programa cuando ya no la necesitaba, seguramente fue un olvido.

COMANDO OLD



Agradeceré se pongan en contacto con Diego Romero Sabio, ya que en la revista nº 3 de abril 1984 página 38 presenta un listado "Comando Old" yo con mi C-64 lo he efectuado pero no he obtenido resultados, quizás esté

...SEAMOS PREGU

diseñado para el VIC-20, no lo sé, según la revista al final del artículo "El misterio de los Pokes" dice que para cualquier duda, que le enviemos las cartas a la revista, por eso les digo al principio que se pongan en contacto con él.
RAFAEL ORTIZ HERNANDEZ
802 ESCUADRON S.A.R.
BASE AEREA DE GANDO
(LAS PALMAS)

El listado que aparece en el artículo "Comando Old" sirve indistintamente para el C-64 y el VIC-20, con ese programa cargas en memoria una rutina en código máquina que graba en el datassette otro programa de sólo dos bytes que es el que sirve para recuperar los programas.

Para recuperar un programa debes pulsar la "tecla de reset", y luego hacer LOAD del programa de dos bytes. ¡No hagas run antes de pulsar "stop y restore"! Es mejor que nos llames por teléfono si sigues sin conseguir nada, quizás tengas algún fallo y no puedo imaginar cuál es si no me das más pistas.

SIGLAS MISTERIOSAS... CODIGOS SECRETOS

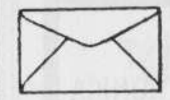


Me gustaría haceros una pregunta:
¿Qué es eso de RTTY y CW? ¿Son programas? ¿Interfaces? ¿Para qué sirven? y otra más: ¿conocéis algún juego de estrategia para el VIC?
MARIO DIEZ
C/COMANDANTE ZORITA, 25
MADRID

RTTY es la abreviatura de Radioteletipo y CW es la de telegrafía por interrupción de portadora, no son ni programas ni interfaces, son modos de transmisión de información utilizados generalmente por aficionados, agencias de noticias, servicios de comunicación militares, etc. Sirven para enviar por radio los mensajes entre dos estaciones.

De momento no tenemos reseña de ningún juego de estrategia, pero tan pronto como la tengamos la publicaremos.

MOSQUITO MOSQUEANTE



Ruego que me digáis la frase 1040 del programa "mosqueando al mosquito", pues no me sale.

Atentamente
ALVARO MARUGAN
AVDA. DEL SEGRE Nº 16
PUIGCERDA (GERONA)

Esa línea se encarga de transferir los datos de los sprites, la línea publicada es correcta, la repetimos por si pareció borrosa en tu ejemplar.

```
1040 FOR I=0 TO 62: READ Z: POKE
832+I,Z:NEXT
```

QUIERO DERROCHAR MEMORIA



Considero a Commodore World, como una buena revista, aunque también es verdad que todavía falta algo de "rodaje".

Peca de exceso de juegos y falta de programas de utilidades técnicas.

Yo tengo un VIC-20 con la máxima configuración (32k) y para almacenaje un cassette.

Y ahora les realizo mis dos preguntas:
¿Por qué todavía no existen programas para sus 32k?

Cuando es un ordenador que todavía le queda mucho por potenciarse.

Y la segunda pregunta es la siguiente:
¿Existe algún programa u orden por la cual se pueda aumentar la velocidad de grabación de carga y grabación en cassette?
GUILLERMO LOPEZ HIDALGO
C/ENTRE ARROYOS, 29
MADRID.

Programas para el VIC-20 con 32K existen, la mayoría de los programas de Bases de Datos suelen necesitar mucha memoria, sobre todo si no tienes disco, también los de contabilidad, proceso de texto, etc., todos estos programas limitan su número de registros o datos en función de la memoria disponible, y aunque en la propaganda te dicen que son para el VIC+8K o +16K, permiten trabajar con más datos si dispones de más RAM.

Existen varios programas para acelerar la velocidad de grabación y carga de cassette, entre ellos están el Fast Turbo y el Arrow. En la revista han aparecido anunciados algunos.

¡BUSCO EL MAPA!



Aprovecho para felicitarles por la revista que es **muy buena** y da un nivel bastante superior a otras publicaciones españolas sobre el tema ¡que no decaiga!

También me gustaría preguntarles (tengo mi Commodore-64 desde hace 1 semana) acerca de los mapas de memoria y el poder del Poke y peek, creía que estas instrucciones tan sólo direccionaban memoria RAM, sin embargo he visto que pueden direccionarse registros de E/S ¿Dónde puedo conseguir un mapa de memoria claro y completo? ¿Quizá la guía del programador? (Todavía no la tengo) ¿Donde comienza la RAM que puedo utilizar libremente?

En el nº 4 de la revista se comentan un poco estos aspectos sin embargo no termino de tenerlo claro y si me aclarasen estas dudas les quedaría agradecido.
CESC SASAL LASAOSA
C/M. FALLA, 24, 2º 3º
BARCELONA

El mapa de memoria completo aparece en la guía de referencia del programador del Commodore 64.

Normalmente la RAM disponible para el usuario comienza en la dirección $0801, que es la 2049 en decimal.

AUTO-RUN



Poseo un Commodore 64 y deseo saber si es posible que como ocurre en otros microcomputadores, un programa que está cargado "arranque" de modo automático.

Muchas gracias por adelantado, un afectuoso saludo.
ANGEL JIMENEZ ROMERO
C/RAMIREZ ARELLANO, 6
CORDOBA - TELEF.: 479 750

Si deseas que el programa arranque sólo después de cargarse, debes cargarlo pulsando las teclas Run/Stop y Shift, esto equivale a teclear LOAD y RUN.

Si lo que quieres es que arranque aunque tú tecleas sólo LOAD, deberás interceptar el sistema operativo del microordenador tal y como lo hacía el programa "comando OLD".

¡QUIERO QUITAR LAS PROTECCIONES!



Me he comprado un programa en cinta que he querido listarlo, pero que no ha sido posible ya que al grabarlo se ejecuta automáticamente y sin hacer caso a las teclas "Run/Stop y Restore".

Me gustaría me dijeran si hay alguna forma de listarlo o de grabarlo en otro cassette.

También hace unos días que al leer en una cinta me sale error. El error me sale en 2 programas, en los otros no me sale error.

El error de esos dos programas puede ser debido a que hayan sido mal grabados o a que esté estropeada ese trozo de cinta.

Uno de los 2 programas hace unos días que iba perfectamente bien ¿Dónde puede estar el error?

Sin nada más que preguntar se despide atentamente.
JOSE MARSA MALLOL
PRATS Y ROQUE, 32
BARCELONA

La mayoría de los programas que puedes encontrar en el mercado se encuentran pro-

NTONES.

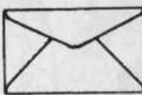


tegidos, esto lo hacen las casas para evitar las copias "piratas". Por eso no responden las teclas de STOP y RESTORE, tampoco puedes listarlo por la misma protección. No te podemos decir cómo desprotegerlo porque las protecciones pueden ser de muchos tipos.

Los errores de cinta pueden ser causados por pequeño desajuste en el azimut de la cabeza, quizás el sello de laca esté desprendido y el tornillo puede haber girado un poco, alterando la posición de la cabeza lectora tan poco que es suficiente para que algunos programas grabados con otro ángulo no entren pero no lo suficiente para que no entre ninguno.

Puedes comprobar si la laca del tornillo se ha desprendido, retocar un poco el ajuste (lo mejor es hacerlo con instrumental, pero por tanteo también es posible), y volver a precintarlo con una gota de laca de uñas.

METIMOS LA PATA



Al querer escribir el programa Aracnida en la línea 60 pone PRINT "(ERROR)"

También ponen ERROR entre corchetes en el programa MAT+DET LINEAS 50, 120 y 50, 140

Como los corchetes los usan, según dicen en la clave para interpretar los listados, para indicar qué secuencia de teclas hay que usar. Por favor indiquen a qué equivale ese error entre corchetes.

Así mismo el programa sobre cómo volver a definir teclas de funciones (Nº 5, pág. 12). No cabrá en la memoria del C-64. Se le acabó la memoria cuando faltaban dos líneas de DATAS:

Y ya de paso corrijan un pequeño error del programa Artillería para el C-64 de la revista nº 2. En la línea 5 pone POKE P 0 y debe poner POKE BO.

JORGE LOZANO
C/SUBIDA DE S. ANDRES, 1
AINZON - ZARAGOZA

La corrección del programa "Arácnida" salió publicada en el número 6, y la de "Mat+Det" en este mismo número.

Lo del programa para utilizar las teclas de función se debe seguramente a algún POKE que hiciste antes de teclear el programa.

En artillería 64 efectivamente tienes razón, debe ser BO en lugar de P0.

ERRATAS Y OMISIONES



Soy asiduo lector de Commodore World, y mi forma de adquirirlo es comprándolo en quioscos o tiendas especializadas, en vez de la consabida suscripción. Esto es debido a que desconfío plenamente de estos nervios por correo, ya que son múltiples los robos de grandes sobres, a medio introducir, de los buzones domiciliarios.

Sólo quiero hacerles un par de sencillas preguntas.

1) en su nº 4 del mes de mayo no especifican para qué aparato es el juego "Viaje lunar", como viene siendo habitual (buena costumbre).

2) En la parte baja del mismo programa, pág. 34, se añaden unas líneas nº 10, 20 y 30, ¿A qué corresponden?

A. A. (BILBAO)

El programa "Viaje Lunar", al igual que el "Mastermind", es de José Vicente López Navarro, y como dice en el recuadro donde aparecen sus señas, es para el VIC-20.

Por un error de imprenta, apareció en primer lugar los títulos y las explicaciones de los dos programas y luego los dos listados, por tanto las líneas que aparecen en la parte baja son en realidad las primeras del listado, y el resto aparece en la página 35.

NOTICIAS COMMODORIANAS

Las fiestas patronales de Palencia con ordenador

La Asociación Palentina de Usuarios de Ordenadores Personales (A.P.U.O.P.) nos comunica que en septiembre tendrá lugar una actividad informática de cara a la ciudad de Palencia.

Dicha actividad consistirá en la instalación en la Plaza Mayor, coincidiendo con las Fiestas Patronales, de una caseta-exposición con diversos equipos y material. Todo ello en un intento de acercar a la ciudad el mundo de la informática y de los Microordenadores.

La instalación será patrocinada por el ayuntamiento que cede la caseta y sus instalaciones y por la firma comercial "La Esfera" que presta los equipos.

Concursos de Ajedrez en Badajoz

Con motivo de las fiestas de San Juan se celebró en Badajoz, en el Parque de Castelar, un concurso de ajedrez para niños. Participaron más de 100 niños de 7 a 14 años, utilizano equipos VIC-20 con el programa Sargon II. Veintiséis niños ganaron al ordenador, 69 salieron perdiendo, y 7 lograron empatar. La Emisión Radio 80 tuvo en el lugar una Unidad Movil que iba retransmitiendo cómo se sucedían las jugadas, ilustranso las anécdotas que se producían con entrevistas a los concursantes y asesores del juego allí presente.

Finalmente, se sorteó un VIC-20 y el programa Sargon II, obsequio del ayuntamiento de Badajoz y de "Control y Sistemas", resultando ganador el niño Javier Millán Núñez, de 12 años de edad.




Las Aventuras de RAMI ROMO y Kbit Kbita
Mi querido LORD...
por ERRAZQUIN
OS ACORDAIS DE AQUELLO QUE ME PEDISTEIS... ¿NO? SI, HOMBRE.. SI.. ERA ACERCA DE COMO SE PODIAN COMUNICAR ENTRE SI DOS ORDENADORES


VOY A LLAMAR A UN AMIGO MIO.. UN LORD INGLÉS AUTENTICO.. EL QUE MAS SABE DE LAS COMUNICACIONES


OH... YESSS... HUMM ¡GOOD! ..¡PERO YA SABES.. HAN DE SER MUY PACIFICOS! ¡BYE, BYE!


VAMOS A IR A VERLE PERO POR FAVOR NO REVOLVAIS NI ROMPAIS NADA DE SU CASA.. ESO LE ENFADARIA MUCHO
PROMETIDO


¡BIENN...! NOS LO VAMOS A PASAR EN GRANDE..
ESTO VA A SER ESTUPENDO... CONOCER A UN LORD!
NO SE YO SI VA A SER UNA BUENA IDEA LLEVARLOS
SI


¡RECORDAD.. NO TOQUEIS NADA Y NO ROMPAIS NADAAAA! ¿VALE?
TOC TOC


TENEMOS UNA CITA CON LORD PING PONG
AGUARDE UN MOMENTO POR FAVOR


PASAD Y SENTAROS MIENTRAS VIENE EL LORD
YUPPI


CRUNCH


AHHH... HABEIS ROTO UNA ESTATUA... COMO SE ENTERE EL LORD...
TENDREMOS QUE ESCONDER LOS TROZOS


¡RAPIDO... METEDLOS BAJO ESTA ALFOMBRA!
NO HACEIS MAS QUE CREAR PROBLEMAS

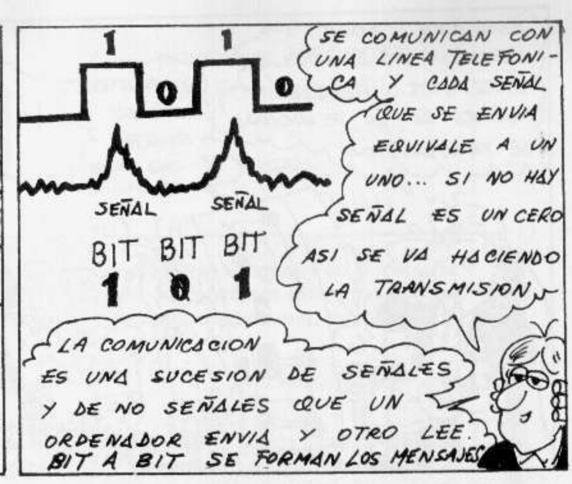

MI QUERIDO AMIGO CHIP CUANTO TIEMPO SIN VERNOS ...¿QUE DESEAS DE MI?


TE VOY A PRESENTAR A UNOS AMIGOS... KBIT, KBITA... Y EL DE LA GORRA ES ROMO .. TODOS QUIEREN SABER COSAS SOBRE COMUNICACIONES
HOLA


BIEN, PUES COMENCEMOS... EL PROBLEMA ES MANDAR DATOS DESDE UN ORDENADOR A OTRO.. DE ESTE MODO PODEMOS ENVIAR INFORMACION DE UN SITIO A OTRO
¡OK!
¿OK?


SU JUSTIFICACION ES CLARA SI YO TENGO INFORMACION EN UN ORDENADOR LO PASO A OTRO POR COMUNICACION .. ASI NO TENGO QUE VOLVER A ESCRIBIRLO
OH
ORD
ORD


CADA CARACTER. CADA BYTE SON UN CONJUNTO DE 8 BITS.. UN BIT ES UN ESTADO CERO O UNO, CADA CARACTER ES UN CONJUNTO, COMO DIJE, DE CEROS Y UNOS. ¡GOOD! EN LA COMUNICACION SE ENVIAN LOS CARACTERES BIT A BIT
A = 8 BITS (1 Ø)
CARACTER
CEROS Y UNOS


1 0 1 0
SEÑAL SEÑAL
BIT BIT BIT
1 0 1
SE COMUNICAN CON UNA LINEA TELEFONICA Y CADA SEÑAL QUE SE ENVIA EQUIVALE A UN UNO... SI NO HAY SEÑAL ES UN CERO ASI SE VA HACIENDO LA TRANSMISION
LA COMUNICACION ES UNA SUCESION DE SEÑALES Y DE NO SEÑALES QUE UN ORDENADOR ENVIA Y OTRO LEE. BIT A BIT SE FORMAN LOS MENSAJES


EL PROCESO SE EJECUTA GRACIAS A UN PROGRAMA QUE SE ENTREGA CON EL ORDENADOR Y ES AUTOMATICO
¿QUE ES ESTO?
AHH


AGGGG ... MY GOD.. ..GOOD!GFFSSMM! MI ESTATUILLA DEL SIGLO XVI HECHA AÑICOS
VOTO A...


HE DE CALMARME... LOS INGLESES NO NOS ENFADAMOS... UN LORD COMO YO EXALTADO... QUE HORROR
AHH HUM


RUEGO ME DISCULPEN... EN REALIDAD ERA UNA ESTATUILLA PEQUEÑA... ¡CONTINUEMOS!
FUE SIN QUERER
SI, SOLO LA EMPUJAMOS UN POQUITO


LA LINEA QUE UNE LOS DOS ORDENADORES PUEDE SER DE VARIOS TIPOS
SI SOLO PUEDE TRANSMITIR EN UN SENTIDO SIN POSIBILIDAD DE QUE LO HAGA EN EL OTRO SE DENOMINA SIMPLEX
CUANDO SE PUEDE ENVIAR INFORMACION DESDE UN PUNTO U OTRO, PERO NO A LA VEZ, SINO CUANDO ACABE EL EMISOR ENTONCES PUEDE RECIBIR. SE LLAMA HALF DUPLEX
Y POR ULTIMO CUANDO SE PUEDE EMITIR Y RECIBIR INFORMACION AL MISMO TIEMPO LA LINEA SE LLAMA COMO FULL DUPLEX
EMISOR
SOLO EN UNA DIRECCION
RECEPTOR
VD. PRIMERO POR FAVOR
NO. VD. HAGA EL FAVOR
EMISOR RECEPTOR
AMBAS DIRECCIONES PERO NO A LA VEZ
EMISOR RECEPTOR
TE ENVIO INFORMACION
Y YO TAMBIEN
EMISOR RECEPTOR
AMBAS DIRECCIONES A LA VEZ
EMISOR RECEPTOR


CADA UNO CONTRATA SU LINEA SEGUN LE CONVENGA DE FORMA PERMANENTE O LIBRE, COMO UN TELEFONO, Y SOLO LE COBRAN CUANDO LA UTILIZA
ASI SE HACE SU RED, BAJO LA MODALIDAD QUE ANTES OS DIJE SIMPLEX, HALF O FULL DUPLEX
CARAZQUIN


PATAPLONG


AYYYY... MI VAJILLA... LA QUE ME REGALÓ LADY COCONUT.. HECHA POLVOOOO... AGGGG


CALMATÉ ...SON UNOS CRIOS ... ADEMAS TU PARA QUE QUIERES UNA VAJILLA TAN HORTERA... SIENTATE Y SIGUENOS CONTANDO COSAS QUE ES MUY INTERESANTE ... ¡PORFA!


ESTA BIEN.. GOOD.. PERO QUE NO SE VUELVA A REPETIR... HABLARÉ AHORA DE LA VELOCIDAD DE TRANSMISION
Y VOSOTROS.. ESTAROS QUIETOS DE UNA VEZ
LA VELOCIDAD SE MIDE EN BAUDIOS, O LO QUE ES LO MISMO EN BITS POR SEGUNDO. LAS VELOCIDADES MAS USUALES OSCILAN DESDE 300 A 9.600 BAUDIOS.
SEGUN LAS NECESIDADES SE ELIGE UNA LINEA CON MAS O MENOS VELOCIDAD EN CADA MOMENTO
YUPI!!!
1 BAUDIO
1 BIT/SEG
CLINCH
ARRFFF GRUMMBLLLL GRRR... ZMSAF
OS VOY A HACER PEDAZOS... VOTO A.. GRUMPPFBBLLL
QUIETO... POR FAVOR.. QUERIDO LORD... VAMOS VAMOS
NOOOO NO ME CALMARE
¿WHAT?
MIRAD UNA MOSCA, UNA MOSCA ENORME
ZZZZZZ
ZZZZZ
DEJADMELA A MI QUE OS SOLUCIONO EL PROBLEMA EN UN SEGUNDO
¡NOOO!
HOP
POF
ZZZFSSSS
CLINC
POM
CLAS
YA NO PUEDO AGUANTARLO MAAAAAAS
BUENO... EJEM... NOSOTROS NOS VAMOS QUE SE HACE TARDE... EJEM.. ADIOS LORD
NO SE POR QUE SE ENFADA TANTO SI MATÉ LA MOSCA ¿NO?
SUELTAMEEE CHIP VOY A HACERLES PICADILLOOOO
ERRAZQUIN

MARKETCLUB

Servicio gratuito para nuestros lectores particulares. Empresas 300 ptas., por línea.
De no especificar duración, los anuncios serán publicados durante 1 mes.

MERCADILLO

• Vendo un ordenador Commodore 4000 y otro 3.000. Buen estado. Con programas. Teléfono: 402 38 89 / 401 21 05 ***(Ref. M-70).***

• Vendo adaptador de alta resolución gráfica para vídeo genie sistems. Convierte la pantalla en 384×192 puntos, añade 10 comandos basic para gráficos alta resolución, vectores, punto a punto, superficies, relleno o vaciado de superficies, etc. Fácil instalación al ordenador a través del conector trasero. Incluye instrucciones de montaje y uso con programas de ejemplo. Convierte al genie EG-3100 en un nuevo ordenador. Precio 25.000 ptas. Interesados llamar al Teléf. (93) 762 44 21 en horas de comida o cena. Ferran Vallespi. Apartado 119. Pineda de Mar. Barcelona. ***(Ref. M-71).***

• Vendo VIC-20 (29.000 ptas.) Superexpander (6.000 ptas.) Cartucho de ayuda al programador (5.000 ptas) y placa expansora 4 cartuchos (7.000 ptas.) por la compra en conjunto regalo "guía de referencia" (2.500 ptas) y "VIC-20 guía del usuario" (2.300). José Antonio Serrano Bergali. C/Avda. de Kansas City, 32. Teléf. (954) 57 33 08 41007 Sevilla. ***(Ref. M-72).***

• Vendo por 32.000 ptas. ZX Spectrum 16K, a estrenar, obtenido en junio en un concurso. Francisco Sáez. C/Virgen del Rosario, nº 28. Alcantarilla (Murcia). Tfno.: (968) 80 23 64. ***(Ref. M-73)***

• Vendo Cassettes de juegos para Commodore 64 por cambio de modelo. Los juegos son: 'Camelos mutantes, frogger, gridrunner y comecocos a 1.000 pesetas cada uno y superbasic (añade 35 instrucciones) a 2.000 pesetas. Mis señas son: Bernardo Berrocal. C/Juan XXIII nº 24., bajos. San Cugat del Vallés (Barcelona). ***(Ref. M-74).***

• Vendo VIC-20 (Comprado en 1983) con ampliación de 16K, cartucho de ajedrez, 1ª y 2ª parte del curso de basic, un libro con treinta programas, cartucho de juegos, cuatro cintas de juegos compradas y una cinta con un montón de juegos y programas. Todo sólo por 37.000 ptas.
José Arturo Guillén Valen, Marina Española, Edificio Azabache, 14-C. Zaragoza-6. ***(Ref. M-75).***

• Tengo un Commodore 64 y deseo intercambiar programas y juegos. Interesados dirigirse a Oriol Fructuoso Recosens C/Sant Eduard 15, 2º 1º Sant Cugat del Valles. Barcelona. Teléfono: (93) 674 51 62. También tendría interés en contactar con un club de usuarios próximo. ***(Ref. M-76)***

• Intercambio programas para el C-64 y VIC-20. Tengo más de 40 programas diferentes. Si me escribes te mandaré un programa regalo. Ignacio Pascual Marugan C/Urgel 288 8-E 08036 Barcelona. ***(Ref. M-77).***

• Vendo VIC-20 (Enero-83), Cassette C2N, y con ellos cartucho Ajedrez, diversos programas en cassette, curso introducción al Basic (Parte 1), Guía de referencia al programador, Colección Club Commodore y Commodore World incluido este año. Todo ello por 45.000 ptas. Llamar al (93) 335 59 13 y preguntar por Antonio, de 13 a 15 h. o a partir de 22 h. Antonio Luque Luz.C/Francia, 56, 10º-2ª, Hospitalet (Barcelona). ***(Ref. M-78).***

• Vendo ZX-81 sin estrenar (lo conseguí en un concurso) y una cinta con diversos programas por 10.000 ptas. Francisco Saez Soto. C/Virgen del Rosario, 28 Alcantarilla (Murcia). Teléf.: (968) 80 23 64 ***(Ref. M-79).***

• Lamento tener que vender unos vídeojuegos Binatone (color) con 4 cartuchos de juegos (comprado todo en el 82) o cambiarlos por impresora para VIC-20 abonando diferencia a convenir. Llamar horas comida o de 21 a 24 h. al (93) 753 11 36. Preguntar por Sergio ***(Ref. M-80).***

• Poseo cartucho monitor lenguaje máquina, y me gustaría cambiarlo por otro, a ser posible de ampliación Ram (mínimo 8K). Abonaría diferencia con programas de juegos (cinta comercial). También admitiría cartucho de juegos. Ah! para VIC-20. José Campos Mtnez. Avda. 26 de julio nº 39. Tuy (Pontevedra) ***(Ref. M-81).***

• Vendo lote de 6 cassettes nuevas compradas en EE.UU. para VIC-20. Precio 2.000 ptas. Vendo curso de Introducción al lenguaje Basic Parte I para VIC-20 (libro y dos cassettes) Precio: 1.750 ptas. Vendo juego "TRON" para C-64 cassette nueva. Precio 1.600 pesetas. Javier. Teléfono (91) 244 42 87. Llamar fines de semana ***(Ref. M-82).***

• Intercambio información y programas de todo tipo, en disco o cassette, para Commodore 64. Interesados dirigirse a: Antonio Fernández Anglada, C/Moralzarzal, 94 (bajo c). Teléfono 734 27. 34. Madrid-34. ***(Ref. M-83).***

• Compro impresora, enviar marca, modelo fecha de compra y precio a: José Manuel. Apartado 2144. La Coruña ***(Ref. M-84).***

• Cambio cassettes por razones industriales, entrego el modelo nuevo con total garantía a cambio del antiguo, o compro el antiguo. Teléf. (93) 230 48 05 ***(Ref. M-85).***

CLUBS

• Club de amigos del VIC-20 y Commodore 64. Ramón O. Serna Soler. Foto Estudio 2. Plaza de Sombrereros, 2. **PALMA DE MALLORCA**. Tel.: 21 31 62. ***(Ref. C-1).***

• Desea información sobre clubs en existencia y gente que quiera formar uno. Agustín Quevedo Velasco. C/ Oña, 55-4º B. **MADRID-34**. Tel.: 202 94 28. ***(Ref. C-2).***

• Desea información sobre clubs en existencia y gente que quiera formar uno. Pablo Nistal Alonso. C/ Empecinado, 46, 6º B. **MOSTOLES**. Tel.: 645 21 70. ***(Ref. C-3).***

• Club de programación Alaiz, en Pamplona. Está dirigido a estudiantes de B.U.P. y C.O.U. No tiene, ningún fin lucrativo. Equipos COMMODORE 64. VIC-20 y diversos periféricos. Pza. Monasterio Santa Gemma, s/n. Tels.: 254 480-257 704. **PAMPLONA**. ***(Ref. C-4).***

• Club de Commodore de **Albacete**. Fer-

nando Martínez Guerrero. La Roda, 39. 5º D. **ALBACETE.** ***(Ref. C-5).***

• Círculo de Durensan Vigués. **VIGO**. C/ Venezuela, 48. Entrechan. Tels.: (986) 410 683/422 519.. ***(Ref. C-6).***

• Un grupo de usuarios de la Comunidad Autónoma de Murcia intenta crear un Club de Usuarios de VIC-20 y C-64 en esta región, sin fines lucrativos. Interesados llamar a los teléfonos (968) 80 23 64 (Mañanas, preguntar por Paco); (968) 26 72 62. (Tardes, preguntar por Fernando). Francisco Sáez Soto. C/ Virgen del Rosario, 28. **ALCANTARILLA**. Murcia. ***(Ref. C-7).***

• Deseo contactar con interesados en formar un club de microordenadores en Barcelona e intercambiar experiencia. Domenec Garrofe. C/ Aragón, 575, 4º 2º. **BARCELONA-26**. Tel.: 348 00 39. ***(Ref. C-8).***

• Interesados en la creación de Club de Usuarios en Girona escribir a "Apartado de Correos nº 580" de **GIRONA**. ***(Ref. C-9).***

• Xabier Prat i Burdo que vive en c/ Rabal Sta. Eulalia s/n. de Sta. María Dólo **(BAGES)** Barcelona, quisiera saber si existe algún Club Commodore cerca de Manresa y solicta información sobre todo lo relacionado con Modems. ***(Ref. C-10).***

• Os rogaría me comunicárais si existe algún Club Commodore en mi ciudad **(GUADALAJARA)** o en **ALCALA DE HENARES** (donde trabajo). Miguel Angel Capeña Alonso. C/ Tabernero, 16-3º I. Tel.: (91) 22 67 57. ***(Ref. C-11)***

• Desearía información sobre la existencia de algún Club Commodore en Somaén o sus cercanías. Carlos Cabrera. Camino de las Eras, s/n. **SOMAEN** (Soria). ***(Ref. C-12).***

• Asociación Manchega de Usuarios de CBM-64 (A.M.U.C.-64). C/ Blasco de Garay, 10. **ALBACETE**. Esta asociación persigue extender el uso racional de la informática, concretándose de momento en el ordenador Commodore CBM-64. ***(Ref. C-13).***

• Club MICRO-ESPLAI. Estamos en Gran Via, 682, Pral. (Entre Bruch y Gerona) de **BARCELONA**. Todos los amigos que tengáis más de 15 años y menos de 100 podéis participar en esta experiencia. Y si sois más jóvenes, venid también, ya nos las ingeniaremos para organizar actividades interesantes.

Para informaros, los viernes de 6 a 8 tarde o los sábados de 9 a 11 de la mañana. ***(Ref. C-14).***

• Queremos formar un Club Commodore en Barcelona. José. Tel.: 329 86 31. C/ Aurora, 19-4º 1.ª **BARCELONA**. ***(Ref. C-15).***

• Me dirijo a ustedes por un hecho reciente, que es la disolución del único club de usuarios en Barcelona con sede en ELEKTROCOMPUTER. Quiero decirles que su revista es un medio para crear un nuevo Club, por eso pido que si ustedes lo creen conveniente, publiquen esta carta. Eduardo Macip Fite. C/ Horno de la Villa, nº 4. ***MONTBLANC***. Tarragona. ***(Ref. C-16)***

• Quisiera saber si existe algún club commodore en Sevilla. Francisco Javier Recio Lamata. C/ Fernán Caballero, 22. **SEVILLA-1.** ***(Ref. C-17).***

• Agradecería mucho a todo el que pudiera informarme sobre la creación del club commodore en Girona. Daniel Bassas. **GERONA** capital. Plaza. Marqués de Camps, 16, 3º, 2.ª. Tel.: 21 27 26. ***(Ref. C-18).***

• Club Ensenada de Santander: Commodore 64 y VIC-20 y diversos periféricos. Su actividad se centra en el intercambio de programas y experiencias, perfeccionamiento de lenguajes, uso del logo, etc. Club Ensenada. C/ Sta. Lucia, 1, 1º. Tel.: 22 59 92. ***(Ref. C-19).***

• Estoy formando un club de intercambio de programas (en cinta) para VIC-20. Jaime Vidal Forteza. C/ Sor Damiana, 16-1º. S. Francisco (Pil. lari). **PALMA DE MALLORCA.** ***(Ref. C-20).***

• Me interesaría conocer si hay algún club de amigos del VIC-20 cerca de mi ciudad. Agustín Ruiz Castrillo. C/ Gregorio Solabarrieta, 49. 4º izda. **MIRANDA DE EBRO** (Burgos). ***(Ref. C-21).***

• Estamos formando un Club de Usuarios en el centro de Madrid. Cualquier interesado que se ponga en contacto con nosotros en el apartado 427 de Madrid, dirigiéndose a Rafael (EB4XD). Teléf. 276 66 94 (Dejar teléfono. Contestador). ***(Ref. C-22)***

• Me gustaría entrar en contacto con usuarios del CBM-64, para intercambio de programas e ideas, así como para trabajar en grupo".

Interesados ponerse en contacto con: Antonio Iglesias Arias. C/San Isidro Labrador, 16, 1º B. Palma de Mallorca, 5. Baleares. Teléf.: 412 728. ***(Ref. C-23).***

• Quisiera que apareciera este aviso: Deseo entrar en contacto con otros poseedores del Commodore 64 para intercambio de programas y técnicas. Angel Jiménez Romero, C/Ramírez Arellano, 6, Córdoba 2. Teléf.: 479 750 ***(Ref. C-24).***

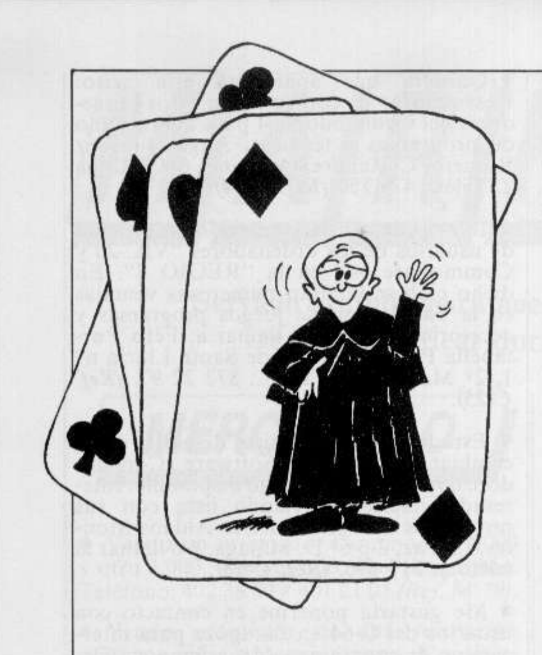
• Interesados en la formación de un club de usuarios de los ordenadores "VIC-20 y Commodore 64" en la "REGIO 7". En dicho club se tendrán numerosas ventajas en la adquisición de juegos programas y accesorios. Escribir o llamar a: Félix Portabella Padro. C/Forn de Santa Llucía nº 1, 2º MANRESA. Teléf.: 872 22 97. ***(Ref. C-25).***

• Estudiante Informática desearía intercambiar todo tipo de Software (Commodore 64). Amplio catálogo disponible. Interesados escribir enviando lista con sus programas a: José Francisco Aldana Montes. La Paz, 2-3 6º D. Málaga, 4 o llamar al teléfono 317 696. ***(Ref. C-26).***

• Me gustaría ponerme en contacto con usuarios del C-64 en Zaragoza para intercambio de programas. Mis señas son: Clemente Pérez Rguez. Avda. de Goya, 2, 1º A. Zaragoza-6. Tfno.: 21 90 96. ***(Ref. C-27).***

BOLSA DE TRABAJO

• Se necesita vendedor de publicidad dinámico con experiencia (Residente en Madrid). Contactar con: COMMODORE WORLD, C/Barquillo, 21-3º Izda. 28004 Madrid. Teléf. 231 23 85/86



VIDEO CASINO

Este juego, que hace uso de los números aleatorios, te puede hacer perder los nervios. Se puede jugar con el VIC-20 ó el C-64

NERVIOS

Este mes, con un juego que es el contrario de como se juega a los dados (es decir, tienes que evitar sacar tu número), vamos a descubrir cómo el VIC-20 y el C-64 eligen números aleatorios.

En el juego "Nervios", tú y tu contrincante intentáis acumular 500 puntos. Cuando toca tu turno, tiras un dado, y continúas tirando todas las veces que te atreves. El número de cada tiro se suma al bote. Sin embargo, si vuelves a sacar la misma cantidad que en tu primera tirada, se resta de tu puntuación la cantidad de puntos de bote.

Puedes seguir tirando, o plantarte en cualquier momento y llevarte el bote. Si te plantas demasiado pronto (porque te falta nervio) el bote que te llevarás no será de mucha cantidad.

Puede llegar un momento en que te pones muy nervioso. Tú tienes 400 puntos, y tu contricante, 480. Hay 90 puntos en el bote. Es mejor tirar, con la posibilidad de llevarte los puntos necesarios para pasar de los 500 y ganar. ¿O es mejor llevarte los 90 puntos y esperar que tu contrincante no se lleve 20 puntos en el siguiente turno? Si te sale mal el tiro, significa que te restan 90 puntos, y te quedas con 310, y la seguridad de perder. ¿Vas cogiendo idea?

"Nervios" es un juego fácil de aprender, ya que las posibilidades de ganar son muy parecidas a las del juego de dados. Si sacas un 12 ó un 2 la primera vez que tiras tienes más posibilidades de llevarte el premio gordo. Podrías acumular 100 ó 200 puntos (incluso más) antes de que vuelva a salir tu primer número. A medida que los puntos se vayan acercando a los números medianos, se aconseja más precaución. Este juego es especialmente divertido, ya que un 12 podría salir tres veces seguidas. Podría ocurrir, juegas tú contra las posibilidades.

La operación del programa es sencilla. Se tiene acceso a una rutina de "tirada de dados" cada vez que sea necesario. La primera tirada se almacena en la variable FR, y cada vez que una tirada posterior sale igual que FR, el programa se bifurca a la rutina "tú pierdes", y resta la cantidad del bote, TT, de tu dinero. Estos valores de dinero se almacenan en un "array" de dos elementos, Cash(1) y Cash(2), cuando el subíndice queda determinado según sea la variable Player (1 ó 2).



TT se incrementa por la cantidad de cada tirada, hasta que te quedes con el bote, o hasta que vuelve a salir tu número original, y TT se resta de tu total. Cuando se finaliza un turno, el programa verifica las líneas 580-600 para ver si el jugador es

VIC-20 ó C-64

el 1 ó el 2, y fija su valor en el contrario. Después de cada tirada, se realiza una verificación, para ver si tú o tu contrincante tiene más de 500 puntos. Si es así, el programa bifurca a una rutina que anuncia el ganador.

Tiradas Aleatorias

Este programa introduce el concepto de RND, la selección de un número aleatorio por el ordenador. Cuando el VIC-20 y el C-64 encuentran la sentencia RND (1), elegirán un número mayor que cero, pero menor que uno. Este podría ser, .562391. .29171, o alguna otra fracción decimal.

Sin embargo, como esto es un juego de dados, necesitamos números enteros de 1-6. Para producir estos números, se multiplica por el mayor número entero que se necesita y se le suma uno. Por ejemplo, RND(1)*6 producirá números reales mayores que cero pero menores que séis. Si a cualquiera de estos números se le añade uno, se consiguen los números aleatorios entre 1+ y 6+. La porción de enteros del número proporciona los números enteros de 1-6.

¿Son los números realmente aleatorios? A decir la verdad no, ya que el ordenador utiliza una fórmula fija (algoritmos) para llegar a una serie de números llamados seudo-aleatorios. Esta serie es muy larga, y el ordenador generalmente empieza en un punto muy distinto en la secuencia cada vez, de forma que los números se repiten pocas veces.

Los programadores avanzados querrán saber que el número trabajado por la sentencia RND (el argumento) afecta el punto inicial de la secuencia. Esto se llama la "semilla". RND(0) generará un número aleatorio que se relaciona con el reloj incorporado del VIC-20 y el C-64. El reloj empieza a contar, a intervalos de 1/60 de segundo, a partir del momento en que se enciende el ordenador.

Si el argumento es menor que cero (RND(-1), por ejemplo), la secuencia de números aleatorios se vuelve a "sembrar" automáticamente. Los argumentos mayores que cero, como RND(1), producirán la misma secuencia de números aleatorios para una determinada "semilla" de número aleatorio.

Realmente las diferencias afectan poco los programas cortos como este. Teclea el siguiente programa corto para ver lo que pasa.

```
10 INPUT "ENTRA ARGUMENTO:";X
20 R=RND(X)
30 PRINT R;
40 GET A$: IF A$=" "GOTO 40
50 GOTO 20
```

Ejecútalo varias veces, introduciendo valores distintos para X, y observa las secuencias. Pulsa cualquier tecla para ver el siguiente número aleatorio. Entre cada ejecución, pulsa la tecla Run/Stop la tecla Restore para asegurar que el ordenador esté totalmente recuperado. ■

VIC-20

```
10 REM NERVIOS
20 REM
30 REM
40 REM
50 REM
60 PRINT"[CLR][2CRSRD]"
70 PRINT"ENTRA[SPC]NOMBRE[SPC]JUGADO
R[SPC]1"
80 INPUT PLAYER$(1)
90 PRINT"ENTRA[SPC]NOMBRE[SPC]JUGADO
R[SPC]2"
100 INPUT PLAYER$(2)
110 PRINT"[CLR][2CRSRD]"
120 PRINTTAB(7)"[RVSON][RED]NERVIOS
[RVSOFF][BLU][2CRSRD]"
130 PRINTTAB(1)"INTENTA[SPC]LLEGAR[SPC]
A[SPC]500"
140 PRINTTAB(2)"PUNTOS[SPC]ANTES[SPC]
QUE"
150 PRINTTAB(2)"TU[SPC]CONTRINCANTE.
"
160 PRINT"PUEDES[SPC]TIRAR[SPC]LOS[SPC]
DADOS[2SPC]HASTA[SPC]QUE[SPC]";
170 PRINTTAB(2)"VUELVAS[SPC]A[4SPC]S
ACAR[SPC]TU[SPC]PRIMERA[4SPC]PUNTUAC
ION[SPC]Y[SPC]PIERDAS,"
180 PRINTTAB(2)"O[SPC]TE[SPC]PLANTAS
[SPC]Y[SPC]TE[6SPC]LLEVAS[SPC]EL[SPC]
BOTE!"
190 PRINT"[CRSRD]"
200 PRINT"[RVSON][GRN][SPC]PULSA[SPC]
CUALQUIER[SPC]TECLA[BLU]"
210 GET A$:IF A$="" GOTO 210
220 PLAYER=1
230 OP=2
240 CASH(1)=200
250 CASH(2)=200
260 GOTO 290
270 ROLL=INT(RND(1)*6)+INT(RND(1)*6)
+2
280 RETURN
290 PRINT"[CLR][2CRSRD]"
300 PRINTTAB(2)"[RVSON][GRN]";PLAYER
$(PLAYER);"[BLU][RVSOFF][SPC]LLEVAS:
[CRSRD]"
310 GOSUB 270
320 FR=ROLL
330 GOSUB 270
340 IF FR=ROLL GOTO 330
350 GOTO 400
360 PRINT"[CLR][2CRSRD]"
370 GOSUB 270
380 PRINTTAB(2)"[RVSON][GRN]";PLAYER
$(PLAYER);"[BLU][RVSOFF][SPC]LLEVAS:
[CRSRD]"
390 IF FR=ROLL GOTO 620
400 IF CASH(1)>500 OR CASH(2)>500 GO
TO 720
410 PRINTTAB(2)"PRIMERA[SPC]TIRADA:"
;FR
420 PRINT"[CRSRD]"
430 PRINTTAB(2)"ULTIMA[2SPC]TIRADA:"
;ROLL
440 PRINT"[CRSRD]"
450 TT=TT+ROLL
460 PRINTTAB(2)"TU[SPC]TOTAL[SPC]ES[SPC]
DE:";CASH(PLAYER);
470 PRINT"[CRSRD]"
480 PRINTTAB(2)"EN[SPC]EL[SPC]BOTE[SPC]
HAY:";TT
490 PRINT"[CRSRD]"
500 PRINT"PUNTOS[SPC]CONTRARIO:";CAS
H(OP)
510 PRINT"[CRSRD]"
520 PRINT"[RVSON][RED]TIRAS[SPC]OTRA
[SPC]VEZ[SPC](S/N)?[SPC][BLU]";
530 GET AN$:IF AN$="" GOTO 530
540 IF AN$="N" THEN CASH(PLAYER)=CAS
H(PLAYER)+TT:GOTO 570
550 IF AN$="S" THEN FOR N=1 TO 100:N
EXT N:GOTO 360
560 GOTO 530
570 TT=0
580 IF PLAYER=1 THENPLAYER=2:OP=1:GO
TO 290
590 OP=2
```

```
600 PLAYER=1
610 GOTO 290
620 PRINT"[CLR][2CRSRD]"
630 PRINTTAB(2)"TU[SPC]PIERDES!"
640 PRINTTAB(2)"TU[SPC]TIENES";
650 CASH(PLAYER)=CASH(PLAYER)-TT
660 PRINT CASH(PLAYER)
670 PRINT"[CRSRD][RVSON][GRN]PULSA[SPC]
CUALQUIER[SPC]TECLA[SPC][BLU]"
680 GET A$:IF A$="" GOTO 680
690 TT=0
700 PRINT"[CLR]"
710 GOTO 570
720 PRINT"[CLR][2CRSRD]"
730 IF CA(1)>CA(2)THEN PRINTTAB(2)PL
$(1);"[SPC]GANA!!":PRINTTAB(2)"[SPC]
CON";CA(1);"$":GOTO 760
740 PRINTTAB(2)PL$(2);"[SPC]GANA!!"
750 PRINT"[SPC]CON";CASH(2);"$"
760 PRINT"[2CRSRD]"
770 PRINTTAB(4)"[RVSON][GRN]JUEGAS[SPC]
OTRA[SPC]VEZ?[BLU]"
780 GET A$:IF A$="" GOTO 780
790 IF A$="S" THEN RUN
```

C-64

```
10 REM
20 REM
30 REM NERVIOS
40 REM
50 REM
60 POKE53281,3
70 PRINT"[CLR][2CRSRD]"
80 PRINTTAB(8)"ENTRA[SPC]EL[SPC]NOMB
RE[SPC]DEL[SPC]JUGADOR[SPC]1"
90 INPUTPL$(1)
100 PRINTTAB(8)"ENTRA[SPC]EL[SPC]NOM
BRE[SPC]DEL[SPC]JUGADOR[SPC]2"
110 INPUTPL$(2)
120 PRINT"[CLR][2CRSRD]"
130 PRINTTAB(15)"[RVSON][RED]NERVIOS
[RVSOFF][BLU][2CRSRD]"
140 PRINTTAB(8)"INTENTA[SPC]ALCANZAR
[SPC]500"
150 PRINTTAB(8)"PUNTOS[SPC]ANTES[SPC]
QUE[SPC]TU[SPC]"
160 PRINTTAB(8)"OPONENTE.[SPC]"
170 PRINTTAB(8)"[CRSRD]TIRAS[SPC]LOS
[SPC]DADOS[SPC]HASTA:"
180 PRINTTAB(8)"[CRSRD]1-[SPC]PLANTA
RTE[8SPC]"
185 PRINTTAB(8)"[CRSRD]2-[SPC]QUE[SPC]
PIERDAS[SPC]POR[2SPC]"
190 PRINTTAB(8)"[2SPC]REPETIR[SPC]EL
[SPC]NUMERO[SPC]"
193 PRINTTAB(8)"[2SPC]DE[SPC]TU[SPC]
PRIMER[SPC]DADO[SPC]"
200 PRINT"[CRSRD]"
210 PRINTTAB(12)"[RVSON][GRN]PULSA[SPC]
UNA[SPC]TECLA[BLU]"
220 GETA$:IFA$=""THEN220
230 PL=1
240 OP=2
250 CA(1)=200
260 CA(2)=200
270 GOTO300
280 RO=INT(RND(1)*6)+INT(RND(1)*6)+2
290 RETURN
300 PRINT"[CLR][2CRSRD]"
310 PRINTTAB(9)"[RVSON][GRN]";PL$(PL
);"[BLU][RVSOFF][SPC]TU[SPC]TURNO.
[2CRSRD]"
320 GOSUB280
330 FR=RO
340 GOSUB280
350 IFFR=ROGOTO340
360 GOTO410
370 PRINT"[CLR][2CRSRD]"
380 GOSUB280
390 PRINTTAB(9)"[RVSON][GRN]";PL$(PL
);"[BLU][RVSON][SPC]TU[SPC]TURNO.
[2CRSRD]"
420 PRINTTAB(9)"PRIMERA[SPC]TIRADA:"
;FR
430 PRINT"[CRSRD]"
440 PRINTTAB(9)"ULTIMA[SPC]TIRADA:";
RO
450 PRINT"[CRSRD]"
460 TT=TT+RO
470 PRINTTAB(9)"TU[SPC]TOTAL[SPC]ES:
";CA(PL)
480 PRINT"[CRSRD]"
490 PRINTTAB(9)"BOTE:";TT
500 PRINT"[CRSRD]"
510 PRINTTAB(9);"PUNTOS[SPC]CONTRARI
O:";CA(OP)
520 PRINT"[CRSRD]"
530 PRINTTAB(12)"[RVSON][RED]OTRA[SPC]
TIRADA?[BLU]"
540 GETAN$:IFAN$=""THEN540
550 IFAN$="N"THENCA(PL)=CA(PL)+TT:GO
TO580
560 IFAN$="S"THENFORN=1TO100:NEXTN:G
OTO370
570 GOTO540
580 TT=0
590 IFPL=1THENPL=2:OP=1:GOTO300
600 OP=2
610 PL=1
620 GOTO300
630 PRINT"[CLR][2CRSRD]"
640 PRINTTAB(9)"TU[SPC]PIERDES![2CRSRD]
"
650 PRINTTAB(9)"TIENES";
660 CA(PL)=CA(PL)-TT
670 PRINTCA(PL)
680 PRINT"[2CRSRD]";TAB(10)"[RVSON][GRN]
PULSA[SPC]UNA[SPC]TECLA[BLU]"
690 GETA$:IFA$=""THEN690
700 TT=0
710 PRINT"[CLR]"
720 GOTO580
730 PRINT"[CLR][2CRSRD]"
740 IFCA(1)>CA(2)THENPRINTTAB(9)PL$(
1);"[SPC]GANA!!":PRINTTAB(9)"CON";CA
(1):GOTO770
750 PRINTTAB(9)PL$(2);"[SPC]GANA!!"
760 PRINT"[SPC]CON"CA(2)
770 PRINTTAB(12)"[RVSON][GRN]JUEGAS[SPC]
OTRA[SPC]VEZ?[BLU]"
780 GET A$:IF A$="" GOTO780
790 IF A$="S" THEN RUN
```

FIN

EXCURSION POR LA MICROPROGRAMACION

Cursillo de Lenguaje Máquina —V—

Por Diego ROMERO

Direccionamiento en Acumulador

Esta forma de direccionamiento utiliza instrucciones de un solo byte, no necesita especificar la dirección donde se encuentra el operando ya se trata del propio acumulador de la CPU. Sólo un tipo de instrucciones de nuestros micros lo utilizan: las de rotación y desplazamiento.

Las instrucciones de desplazamiento equivalen a multiplicar o dividir por dos el contenido del registro A de la CPU, dependiendo de que el desplazamiento sea a izquierda o derecha respectivamente.

Direccionamiento Implícito

Las instrucciones que lo utilizan también son de un solo byte. Pues como su nombre indica, el operando va implícito en la instrucción. Por ejemplo: INX significa INcrementar el registro-índice X, y como podéis imaginar no hace falta especificar nada más.

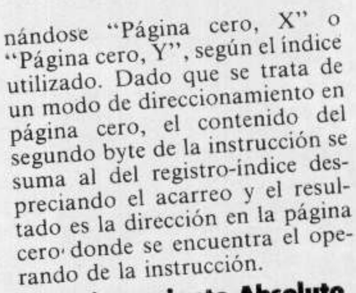
Direccionamiento Indirecto Absoluto

Sólo existe una instrucción que lo utiliza, es la de salto indirecto, los dos bytes que siguen a la instrucción apuntan a una posición de memoria donde se encuentra un vector (otros dos bytes) que es el que realmente indica la dirección a donde debe saltar un programa.

En estos modos de direccionamiento, no he puesto los equivalentes, porque es difícil encontrar algo similar y que no se preste a confusiones en el BASIC implementado por nuestros ordenadores.

Direccionamiento Página Cero Indexado

Esta forma de direccionamiento es utilizada aprovechando los registros-índices X e Y, denominándose "Página cero, X" o "Página cero, Y", según el índice utilizado. Dado que se trata de un modo de direccionamiento en página cero, el contenido del segundo byte de la instrucción se suma al del registro-índice despreciando el acarreo y el resultado es la dirección en la página cero donde se encuentra el operando de la instrucción.



Direccionamiento Absoluto Indexado

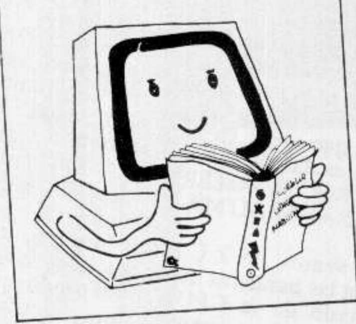
Al igual que el anterior puede ser "Absoluto,X" o "Absoluto,Y", según el índice utilizado. La dirección efectiva del operando se calcula sumando el contenido del índice a la dirección contenida en el segundo y tercer byte de la instrucción. Este modo de direccionamiento es muy utilizado por nuestros equipos para acceder a tablas de mensajes que deben salir a pantalla, valores que deben transferirse, etc., es muy útil cuando debemos manejar en una rutina datos que ocupan posiciones de memoria consecutivas.

Direccionamiento Indexado Indirecto

También se le denomina Preindexado, ya que el segundo byte de la instrucción se suma al contenido del registro índice X despreciando el arrastre, la dirección en página cero a la que señala la suma contiene la parte baja y la siguiente la parte alta de la dirección real del operando, otra denominación utilizada es la de "(Indirecto X)".

Direccionamiento Indirecto Indexado

Denominado también "Post-indexado" o "(Indirecto), Y", el segundo byte de la instrucción señala a una posición en página cero que contiene la parte baja de la dirección, la siguiente posición ▶



EXCURSION POR LA MICROPROGRAMACION

Cursillo de Lenguaje Máquina —V—

(Continuación)

▶ contiene la parte alta, y a este vector se le suma el contenido del registro índice Y, el resultado es la dirección real del operando.

Este ha sido el último modo de direccionamiento, y en el próximo número pasamos a examinar el juego de instrucción de las CPU's. Se trata de 56 instrucciones diferentes, pero se pueden agrupar simplificando y acelerando su estudio para pasar a las aplicaciones prácticas que es lo realmente interesante de la programación en código máquina.

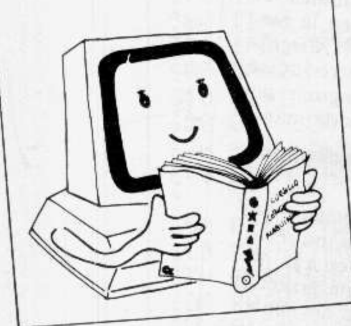
El repertorio de instrucciones lo dividiremos según su función de carga y almacenamiento, suma, resta y comparación, desplazamiento, transferencia, instrucciones lógicas, saltos, etc.

Voy a explicar ahora el contenido de las fichas de códigos mnemónicos que van apareciendo en la revista, para ello nos vamos a fijar en los que aparecen en este mismo número (página anterior).

En primer lugar nos fijamos en unos grupos de siglas, tres letras más grandes que destacan (ej. JMP), ese es el mnemónico de la instrucción de código máquina. Debajo encontramos su significado, que en el ejemplo que hemos tomado es "salta a una nueva dirección". En la siguiente línea vemos lo que hace la instrucción representado con símbolos, en el ejemplo coloca el contenido de la siguiente posición a la del contador de programa (PC) en la parte baja del mismo (PCL), y el siguiente byte en la parte alta (PCH).

La referencia que viene en la siguiente línea, es la del libro "MCS 6500 MICROCOMPUTER FAMILY PROGRAMMING MANUAL".

En otra línea vemos: "N Z C I D V". Estas son las banderas del registro de estado de la CPU, debajo de ellas podemos ver unos



guiones que significan que no han sido alteradas, pero en el caso de que aparezca otro símbolo (ver INX), sí son alteradas al ejecutar esa instrucción en código máquina. Si no sabéis el significado de estas siglas, podéis encontrarlo en las páginas 15 y 16 del número 4 (Mayo).

En el recuadro indica los modos de direccionamiento que se pueden utilizar con esa instrucción, en JMP son dos: absoluto e indirecto absoluto. Al lado vemos el modo en que se escribe la instrucción en ensamblador y los códigos de operación.

El código de operación, es el código hexadecimal que equivale a esa instrucción. Dicho de otro modo, lo que hay que poner en una posición de memoria para que al llegar a ella la CPU ejecute esa instrucción, en nuestro ejemplo si en la posición 2000 ponemos un $4C, al hacer SYS 2000, la CPU saltará a donde le indique los bytes siguientes.

El número de bytes que necesita esa instrucción incluyendo el código de operación y el operando aparece en la siguiente ventana.

Finalmente el número de ciclos sirve para conocer exactamente cuántos microsegundos tardará la CPU en ejecutar la instrucción. Es útil cuando tenemos una rutina en la que es importante el tiempo invertido.

Para saber cuánto tarda en ejecutar una instrucción, debemos multiplicar el número de ciclos por el período del reloj de la CPU. En el VIC 20, la señal de reloj procede de un cristal de cuarzo de 4,433 MHz, y es dividida por cuatro antes de llegar a la CPU, luego el reloj tendrá una frecuencia de 4,433/4 MHz, y su período será 4/4,433 microsegundos (la inversa de la frecuencia), lo que nos da aproximadamente 0,9. Si queremos calcular cuánto tardaríamos en ejecutar la instrucción JMP $2033, multiplicamos el período por el número de ciclos y tenemos: 3*0,9=2,7 microsegundos.

Para dar idea de lo rápido que es el código máquina frente al BASIC, podemos coger el equivalente de esta instrucción, que sería GOTO. La instrucción GOTO tarda unos 16 milisegundos, que son 16.000 microsegundos. ¿Qué pensais de la diferencia? Hasta el próximo número.